


# MELSEC System Q

## Speicherprogrammierbare Steuerungen

## Einsteigerhandbuch

Art.-Nr.: 209092
28082007
Version A

MITSUBISHI ELECTRIC INDUSTRIAL AUTOMATION

# Zu diesem Handbuch

Die in diesem Handbuch vorliegenden Texte, Abbildungen, Diagramme und Beispiele dienen ausschließlich zur Erläuterung der Installation, Bedienung, Programmierung und Anwendung der speicherprogrammierbaren Steuerungen des MELSEC System Q.

Sollten sich Fragen zur Installation und Betrieb der in diesem Handbuch beschriebenen Geräte ergeben, zögern Sie nicht, Ihr zuständiges Verkaufsbüro oder einen Ihrer Vertriebspartner (siehe Umschlagseite) zu kontaktieren.
Aktuelle Informationen sowie Antworten auf häufig gestellte Fragen erhalten Sie über das Internet (www.mitsubishi-automation.de).

Die MITSUBISHI ELECTRIC EUROPE B.V. behält sich vor, jederzeit technische Änderungen dieses Handbuchs ohne besondere Hinweise vorzunehmen.

**Einsteigerhandbuch zu den speicherprogrammierbaren Steuerungen des MELSEC System Q**
**Artikel-Nr.: 209092**

| Version | | | Änderungen / Ergänzungen / Korrekturen |
|---|---|---|---|
| A | 08/2007 | pdp-dk | Erste Ausgabe |

# Sicherheitshinweise

### Zielgruppe

Dieses Handbuch richtet sich ausschließlich an anerkannt ausgebildete Elektrofachkräfte, die mit den Sicherheitsstandards der Automatisierungstechnik vertraut sind. Projektierung, Installation, Inbetriebnahme, Wartung und Prüfung der Geräte dürfen nur von einer anerkannt ausgebildeten Elektrofachkraft, die mit den Sicherheitsstandards der Automatisierungstechnik vertraut ist, ausgeführt werden. Eingriffe in die Hard- und Software unserer Produkte, soweit sie nicht in diesem Handbuch beschrieben sind, dürfen nur durch unser Fachpersonal vorgenommen werden.

### Bestimmungsgemäßer Gebrauch

Die speicherprogrammierbaren Steuerungen des MELSEC System Q sind nur für die Einsatzbereiche vorgesehen, die in diesem Handbuch beschrieben sind. Achten Sie auf die Einhaltung aller im Handbuch angegebenen Kenndaten. Die Produkte wurden unter Beachtung der Sicherheitsnormen entwickelt, gefertigt, geprüft und dokumentiert. Unqualifizierte Eingriffe in die Hard- oder Software bzw. Nichtbeachtung der in diesem Handbuch angegebenen oder am Produkt angebrachten Warnhinweise können zu schweren Personen- oder Sachschäden führen. Es dürfen nur von MITSUBISHI ELECTRIC empfohlene Zusatz- bzw. Erweiterungsgeräte in Verbindung mit den speicherprogrammierbaren Steuerungen des MELSEC System Q verwendet werden.

Jede andere darüber hinausgehende Verwendung oder Benutzung gilt als nicht bestimmungsgemäß.

### Sicherheitsrelevante Vorschriften

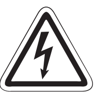
Bei der Projektierung, Installation, Inbetriebnahme, Wartung und Prüfung der Geräte müssen die für den spezifischen Einsatzfall gültigen Sicherheits- und Unfallverhütungsvorschriften beachtet werden. Es müssen besonders folgende Vorschriften (ohne Anspruch auf Vollständigkeit) beachten werden:

- VDE-Vorschriften
  - VDE 0100
    Bestimmungen für das Errichten von Starkstromanlagen mit einer Nennspannung bis 1000V
  - VDE 0105
    Betrieb von Starkstromanlagen
  - VDE 0113
    Elektrische Anlagen mit elektronischen Betriebsmitteln
  - VDE 0160
    Ausrüstung von Starkstromanlagen und elektrischen Betriebsmitteln
  - VDE 0550/0551
    Bestimmungen für Transformatoren
  - VDE 0700
    Sicherheit elektrischer Geräte für den Hausgebrauch und ähnliche Zwecke
  - VDE 0860
    Sicherheitsbestimmungen für netzbetriebene elektronische Geräte und deren Zubehör für den Hausgebrauch und ähnliche Zwecke.
- Brandverhütungsvorschriften

- Unfallverhütungsvorschrift
  - VBG Nr.4
    Elektrische Anlagen und Betriebsmittel

**Gefahrenhinweise**

Die einzelnen Hinweise haben folgende Bedeutung:



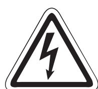
**GEFAHR:**
***Bedeutet, dass eine Gefahr für das Leben und die Gesundheit des Anwenders besteht, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.***



**ACHTUNG:**
***Bedeutet eine Warnung vor möglichen Beschädigungen des Gerätes oder anderen Sachwerten, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.***

**Allgemeine Gefahrenhinweise und Sicherheitsvorkehrungen**

Die folgenden Gefahrenhinweise sind als generelle Richtlinie für den Umgang mit der SPS in Verbindung mit anderen Geräten zu verstehen. Diese Hinweise müssen Sie bei der Projektierung, Installation und Betrieb einer Steuerungsanlage unbedingt beachten.



***GEFAHR***

- ***Die im spezifischen Einsatzfall geltenden Sicherheits- und Unfallverhütungsvorschriften sind zu beachten. Der Einbau, die Verdrahtung und das Öffnen der Baugruppen, Bauteile und Geräte müssen im spannungslosen Zustand erfolgen.***
- ***Baugruppen, Bauteile und Geräte müssen in einem berührungssicheren Gehäuse mit einer bestimmungsgemäßen Abdeckung und Schutzeinrichtung installiert werden.***
- ***Bei Geräten mit einem ortsfesten Netzanschluss müssen ein allpoliger Netztrennschalter und eine Sicherung in die Gebäudeinstallation eingebaut werden.***
- ***Überprüfen Sie spannungsführende Kabel und Leitungen, mit denen die Geräte verbunden sind, regelmäßig auf Isolationsfehler oder Bruchstellen. Bei Feststellung eines Fehlers in der Verkabelung müssen Sie die Geräte und die Verkabelung sofort spannungslos schalten und die defekte Verkabelung ersetzen.***
- ***Überprüfen Sie vor der Inbetriebnahme, ob der zulässige Netzspannungsbereich mit der örtlichen Netzspannung übereinstimmt.***
- ***Damit ein Leitungs- oder Aderbruch auf der Signalseite nicht zu undefinierten Zuständen führen kann, sind entsprechende Sicherheitsvorkehrungen zu treffen.***
- ***Treffen Sie die erforderlichen Vorkehrungen, um nach Spannungseinbrüchen und -ausfällen ein unterbrochenes Programm ordnungsgemäß wieder aufnehmen zu können. Dabei dürfen auch kurzzeitig keine gefährlichen Betriebszustände auftreten.***
- ***Fehlerstrom-Schutzeinrichtungen nach DIN VDE 0641 Teil 1-3 sind als alleiniger Schutz bei indirekten Berührungen in Verbindung mit speicherprogrammierbaren Steuerungen nicht ausreichend. Hierfür sind zusätzliche bzw. andere Schutzmaßnahmen zu ergreifen.***
- ***NOT-AUS-Einrichtungen gemäß EN60204/IEC 204 VDE 0113 müssen in allen Betriebsarten der SPS wirksam bleiben. Ein Entriegeln der NOT-AUS-Einrichtung darf keinen unkontrollierten oder undefinierten Wiederanlauf bewirken.***
- ***Damit ein Leitungs- oder Aderbruch auf der Signalseite nicht zu undefinierten Zuständen in der Steuerung führen kann, sind hard- und softwareseitig entsprechende Sicherheitsvorkehrungen zu treffen.***
- ***Beim Einsatz der Module muss stets auf die strikte Einhaltung der Kenndaten für elektrische und physikalische Größen geachtet werden.***

# Inhaltsverzeichnis

## 5 Operanden im Detail



## 6 Programmierung für Fortgeschrittene

# 1 Einleitung

## 1.1 Dieses Handbuch...

..soll Ihnen die ersten Schritte beim Umgang mit den speicherprogrammierbaren Steuerungen des MELSEC System Q erleichtern. Es richtet sich besonders an Anwender, die noch keine Erfahrung bei der Programmierung von speicherprogrammierbaren Steuerungen (SPS) gesammelt haben.

Aber auch Programmierern, die bisher mit Steuerungen anderer Hersteller gearbeitet haben, kann dieses Handbuch den „Umstieg“ zum MELSEC System Q.

## 1.2 Weitere Informationen...

...und detaillierte Beschreibungen zu den einzelnen Geräten entnehmen Sie bitte den Bedienungs- oder Installationsanleitungen der einzelnen Module.

Der Technische Katalog MELSEC System Q, Art.-Nr. 136729, gibt Ihnen einen Überblick über die Steuerungen des MELSEC System Q. Er informiert außerdem über Sondermodule und das lieferbare Zubehör.

Die vielfältigen Kommunikationsmöglichkeiten über Mitsubishi- und offene Netzwerke, wie Ethernet oder PROFIBUS, sind im technischen Katalog Netzwerke (Art.-Nr. 136728) beschrieben.

Das Hardware-Handbuch zum MELSEC System Q (Art.-Nr. 141683) hilft Ihnen bei der Planung, Installation und Inbetriebnahme Ihrer SPS.

Die ersten Schritte mit der Programmier-Software erleichtern Ihnen das Einsteigerhandbuch (Art.-Nr. 43594) und das Benutzerhandbuch zum GX IEC Developer (Art.-Nr. 43595).

Eine ausführliche Beschreibung aller Programmanweisungen finden Sie in der Programmieranleitung zur MELSEC A/Q-Serie und dem MELSEC System Q mit der Artikel-Nr. 87 432. Zusätzlich sind in den Bedienungsanleitungen der Sondermodule fast immer Programmbeispiele angegeben.

**HINWEIS** Handbücher und Kataloge können kostenlos über die Mitsubishi-Homepage (www.mitsubishi-automation.de) bezogen werden.

# 2 Speicherprogrammierte Steuerungen

## 2.1 Was ist eine SPS?

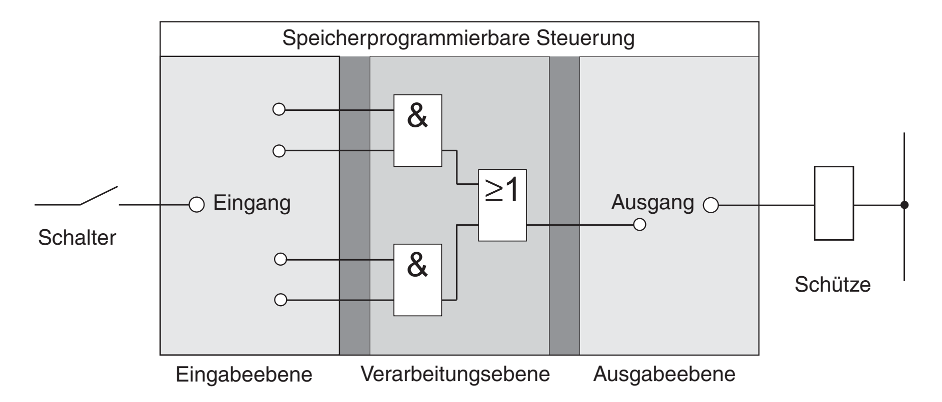
Im Gegensatz zu einer Steuerung, deren Funktion nur durch die Verdrahtung bestimmt wird, wird bei einer speicherprogrammierbaren Steuerung (SPS) die Funktion durch ein Programm festgelegt. Zwar benötigt auch eine SPS zur Verbindung mit der Außenwelt eine Verdrahtung, der Inhalt des Programmspeichers kann aber jederzeit geändert und das Programm an verschiedene Steuerungsaufgaben angepasst werden.

Bei speicherprogrammierbaren Steuerungen werden Daten eingegeben, verarbeitet und die Verarbeitungsergebnisse wieder ausgegeben. Dieser Prozess gliedert sich in:

- eine Eingabeebene,
- eine Verarbeitungsebene

und

- eine Ausgabeebene.




**Eingabeebene**

Die Eingabeebene dient zur Übergabe von Steuersignalen, die von Schaltern, Tastern oder Sensoren stammen, an die Verarbeitungsebene.

Die Signale dieser Bauelemente entstehen im Steuerungsprozess und werden als logischer Zustand den Eingängen zugeführt. Die Eingabeebene übergibt die Signale in aufbereiteter Form der Verarbeitungsebene.

**Verarbeitungsebene**

Die von der Eingabeebene erfassten und aufbereiteten Signale werden in der Verarbeitungsebene durch ein gespeichertes Programm verarbeitet und logisch verknüpft. Der Programmspeicher der Verarbeitungsebene ist frei programmierbar. Eine Änderung des Verarbeitungsablaufs ist jederzeit durch Änderung oder Austausch des gespeicherten Programms möglich.

**Ausgabebene**

Die Resultate, die aus der Verarbeitung der Eingangssignale im Programm entstanden sind, beeinflussen in der Ausgangsebene die an den Ausgängen angeschlossenen Schaltglieder wie z. B. Schütze, Meldeleuchten, Magnetventile usw..

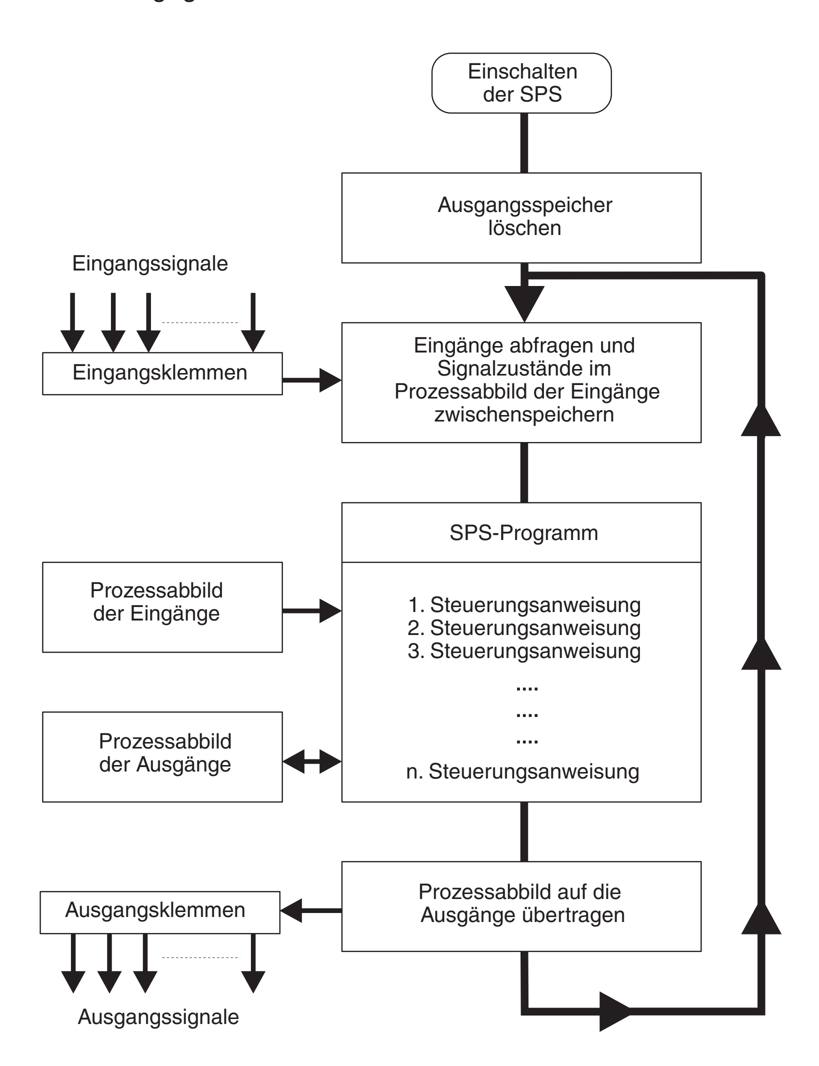
## 2.2 Programmverarbeitung in der SPS

Eine SPS arbeitet nach einem vorgegebenen Programm, das in der Regel außerhalb der Steuerung erstellt, in die Steuerung übertragen und im Programmspeicher abgelegt wird. Für die Programmierung ist es wichtig zu wissen, wie das Programm von der SPS verarbeitet wird.

Das Programm besteht aus einer Folge einzelner Anweisungen, die die Funktion der Steuerung festlegen. Die SPS arbeitet die Steuerungsanweisungen in der programmierten Reihenfolge nacheinander (sequentiell) ab.
Der gesamte Programmdurchlauf wird ständig wiederholt, es findet also ein zyklischer Programmdurchlauf statt. Die für einen Programmdurchlauf benötigte Zeit wird als Programmzykluszeit bezeichnet.

### Prozessabbildverfahren

Bei der Programmbearbeitung in der SPS wird nicht direkt auf die Ein- und Ausgänge, sondern auf ihr Prozessabbild zugegriffen:



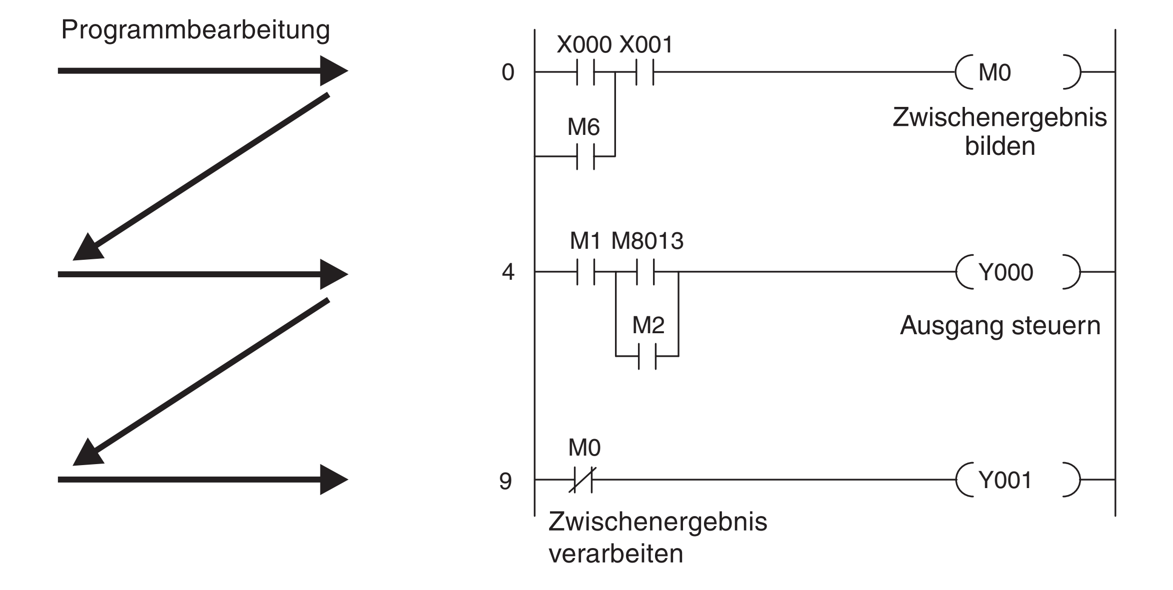

### Prozessabbild der Eingänge

Am Anfang eines Programmzyklusses werden die Signalzustände der Eingänge abgefragt und zwischengespeichert: Es wird ein sogenanntes Prozessabbild der Eingänge angelegt.

### Programmdurchlauf

Während des anschließenden Programmdurchlaufs greift die SPS auf die gespeicherten Eingangszustände im Prozessabbild zu. Signaländerungen an den Eingängen werden daher erst im nächsten Programmzyklus erkannt.

Das Programm wird von oben nach unten, in der Reihenfolge der Eingabe, abgearbeitet. Zwischenergebnisse können noch im selben Programmzyklus verwendet werden




### Prozessabbild der Ausgänge

Verknüpfungsergebnisse, die die Ausgänge betreffen, werden in einem Ausgangszwischenspeicher hinterlegt (Prozessabbild der Ausgänge). Erst am Ende des Programmdurchlaufs werden die Zwischenergebnisse an die Ausgänge übertragen. Im Ausgangszwischenspeicher bleibt das Prozessabbild der Ausgänge bis zum nächsten Überschreiben erhalten. Nach der Wertzuweisung an die Ausgänge wird der Programmzyklus wiederholt.

### Signalverarbeitung in der SPS im Gegensatz zur verbindungsprogrammierten Steuerung

Bei einer verbindungsprogrammierten Steuerung ist das Programm durch die Art der Funktionsglieder und deren Verbindung (Verdrahtung) vorgegeben. Alle Steuerungsvorgänge werden gleichzeitig (parallel) ausgeführt. Jede Änderung der Eingangssignalzustände bewirkt sofort eine Änderung der Ausgangssignalzustände.

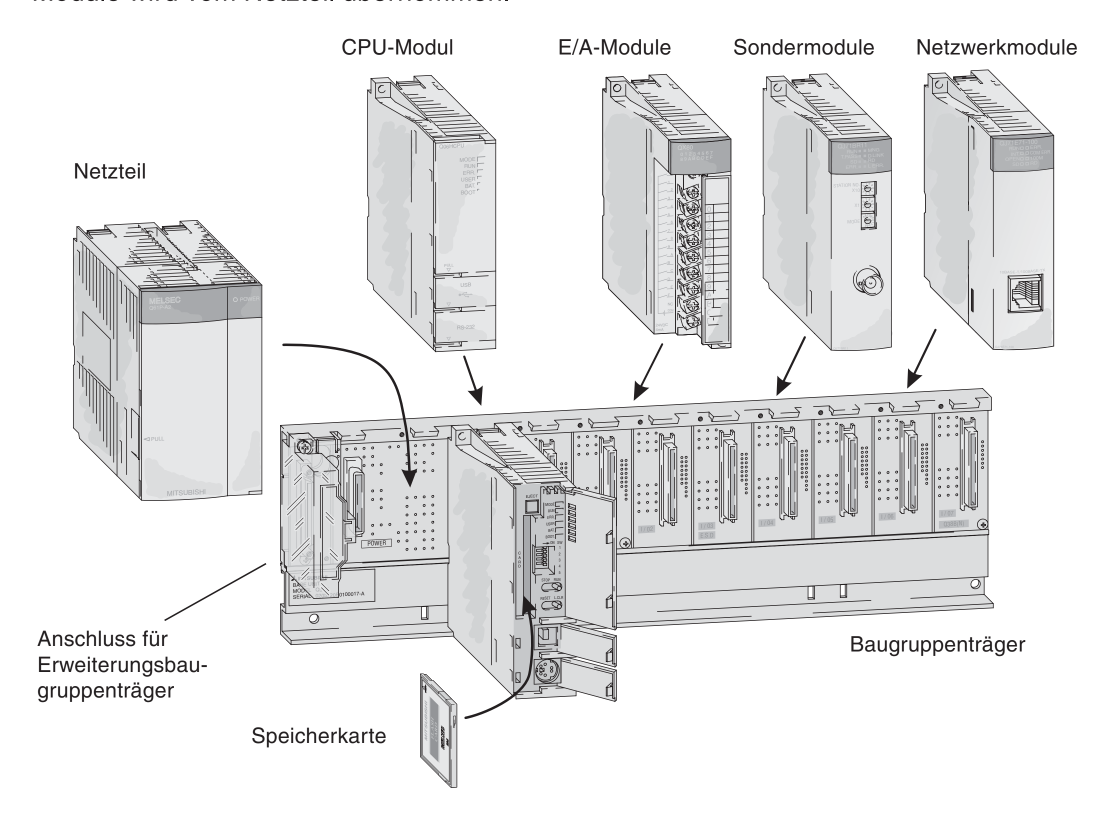
Bei einer SPS kann eine Änderung der Eingangssignalzustände während des Programmdurchlaufs erst wieder beim nächsten Programmzyklus berücksichtigt werden. Dieser Nachteil wird durch kurze Programmzykluszeiten weitgehend wieder ausgeglichen. Die Programmzykluszeit ist abhängig von der Anzahl und der Art der Steuerungsanweisungen.

# 3 Das MELSEC System Q

## 3.1 Systemaufbau

Das MELSEC System Q ist eine leistungsstarke modulare SPS mit Multiprozessortechnik. Modular bedeutet, dass die Systemkonfiguration individuell und optimal an eine Anwendung angepasst werden kann.

Das Kernstück einer SPS wird aus einem Baugruppenträger, einem Netzteil und mindestens einem CPU-Modul gebildet. Die CPU führt die Anweisungen im SPS-Programm aus. Auf dem Baugruppenträger werden – je nach Art der Anwendung – noch weitere Module, wie z. B. Ein- und Ausgabemodule (E/A-Module), montiert. Die Spannungsversorgung der installierten Module wird vom Netzteil übernommen.

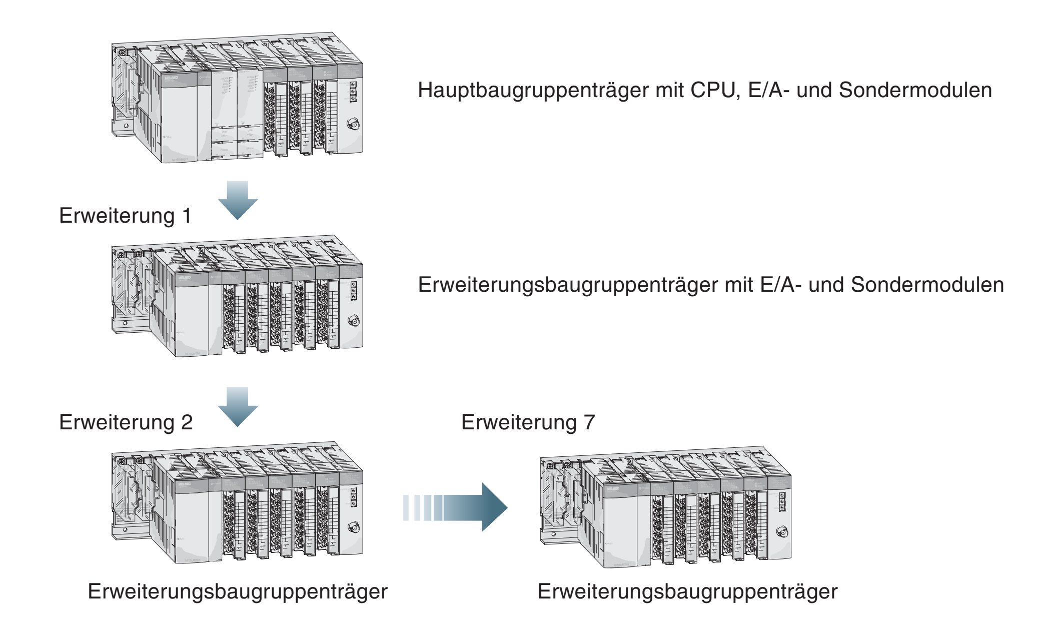



Die Kommunikation zwischen den einzelnen Modulen und der CPU wird über eine interne Busverbindung des Baugruppenträgers abgewickelt.

Der Baugruppenträger, der das CPU-Modul aufnimmt, wird als Hauptbaugruppenträger bezeichnet. Im MELSEC System Q stehen fünf verschiedene Hauptbaugruppenträger mit bis zu 12 Steckplätzen für Module zur Verfügung.

**Erweiterungsmöglichkeiten**

Jeder Hauptbaugruppenträger kann über Erweiterungsbaugruppenträger ergänzt werden, so dass zusätzliche Steckplätze zur Verfügung stehen. Die Baugruppenträger werden durch Erweiterungskabel miteinander verbunden. Diese Verbindungskabel dienen beim Einsatz von Erweiterungsbaugruppenträgern ohne eigenes Netzteil auch zur Spannungsversorgung der dort installierten Module. An einen Hauptbaugruppenträger können bis zu sieben Erweiterungsbaugruppenträger angeschlossen werden. Maximal 64 Module können in Haupt- und Erweiterungsbaugruppenträger eingesetzt werden.

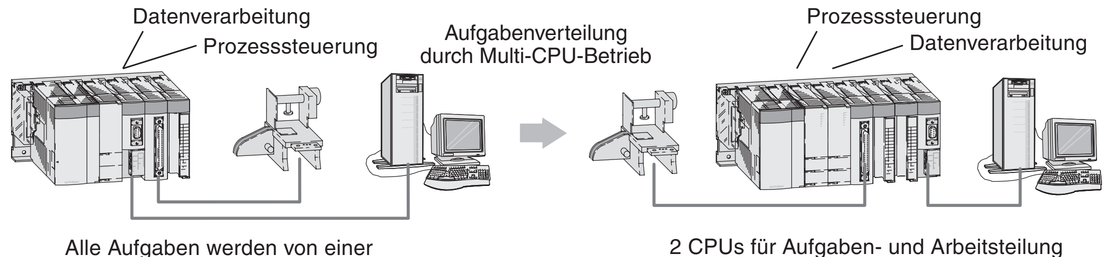


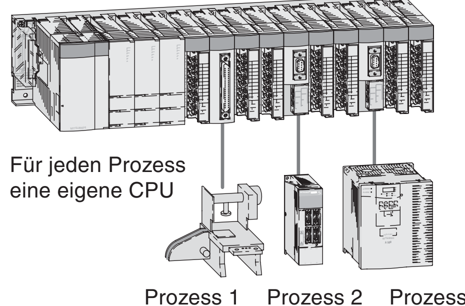

Beim Auswahl des Netzteils muss die Stromaufnahme der Ein- und Ausgangsmodule, der Sondermodule und der peripheren Geräte berücksichtigt werden. Falls erforderlich, muss ein Erweiterungsbaugruppenträger mit einem weiteren Netzteil verwendet werden.

Bei der Verdrahtung von umfangreichen Anlagen oder bei Maschinen mit modularem Aufbau bieten dezentrale Ein- und Ausgänge (E/A-Stationen) Vorteile, die unmittelbar vor Ort angeordnet sind. Dabei können die Verbindungen zwischen den Ein- oder Ausgängen und den Sensoren bzw. den Schaltgliedern kurz gehalten werden. Zur Verbindung zwischen einer dezentralen E/A-Station und dem System mit der SPS-CPU wird nur ein Netzwerkmodul und ein Netzwerkkabel benötigt. Je nach gewähltem CPU-Typ können bis zu 4096 zentrale (auf Haupt- und Erweiterungsbaugruppenträgern) und bis zu 8192 dezentrale Ein-/Ausgänge (über Netzwerke) angesprochen werden.

### Aufgabenverteilung durch Multi-CPU-Betrieb

Durch den Einsatz mehrerer CPU-Module können in einem System Prozesse mit unterschiedlichen Taktzeiten, wie zum Beispiel Ablaufsteuerung und Datenverarbeitung, gleichzeitig gesteuert werden. Prozesssteuerung und Rechenleistung können so auf verschiedene CPUs verteilt werden.

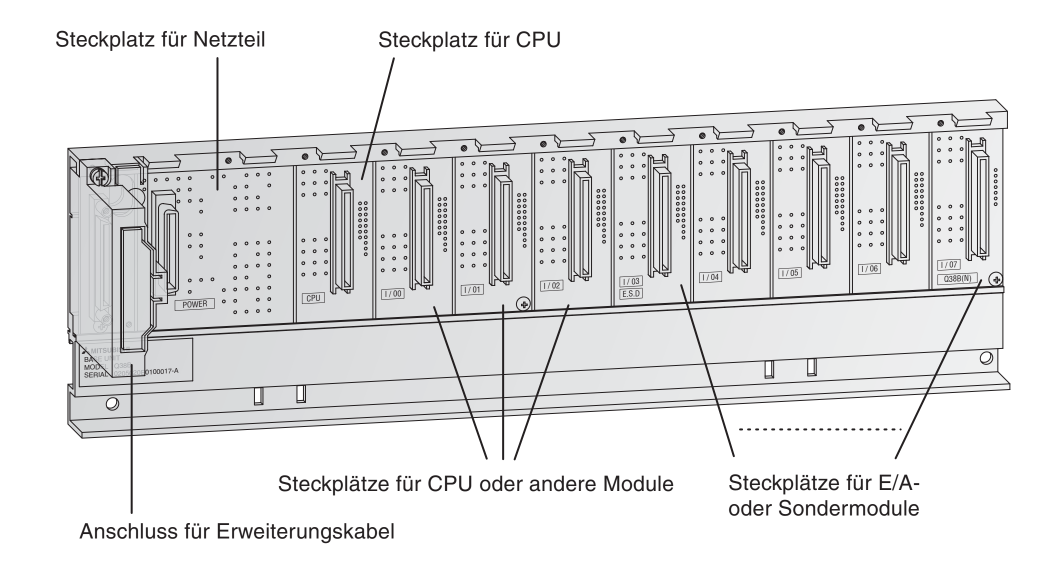



Bei einem komplexen System, bei dem die Leistung einer einzelnen CPU nicht ausreicht, können die Aufgaben auf mehrere CPUs aufgeteilt werden und damit die Leistungsfähigkeit des Systems gesteigert werden.

## 3.2 Baugruppenträger

Die Hauptbaugruppenträger nehmen ein Netzteil, ein oder mehrere CPU-Module und E/A- oder Sondermodule auf. In den Erweiterungsbaugruppenträgern können E/A- und Sondermodule installiert werden. Die Baugruppen werden entweder direkt, z. B. im Schaltschrank, oder mit Hilfe von Adaptern auf einer DIN-Schiene installiert.




In den folgenden Tabellen sind alle erhältlichen Baugruppenträger aufgeführt.

| Merkmal | Hauptbaugruppenträger | | | | |
|---|---|---|---|---|---|
| | Q33B | Q35B | Q38B | Q38RB | Q312B |
| Anzahl der Steckplätze für Netzteile | 1 | 1 | 1 | 2* | 1 |
| Anzahl der Steckplätze für E/A- oder Sondermodule | 3 | 5 | 8 | 8 | 12 |

* Im Hauptbaugruppenträger Q38RB können redundante Netzteile verwendet werden (siehe Abschnitt 2.3.4, 3.3).

| Merkmal | Erweiterungsbaugruppenträger | | | | | | |
|---|---|---|---|---|---|---|---|
| | Q52B | Q55B | Q63B | Q65B | Q68B | Q68RB | Q612B |
| Anzahl der Steckplätze für Netzteile | — | — | 1 | 1 | 1 | 2* | 1 |
| Anzahl der Steckplätze für E/A- oder Sondermodule | 2 | 5 | 3 | 5 | 8 | 8 | 12 |

* Im Erweiterungsbaugruppenträger Q68RB können redundante Netzteile verwendet werden (Abschnitt 2.3.4, 3.3).

### 3.2.1 Erweiterungskabel

Mit den Erweiterungskabeln werden Haupt- und Erweiterungsbaugruppenträger verbunden. Die maximale Länge aller Verbindungskabel darf 13,2 m nicht überschreiten.

| Erweiterungskabel | QC05B | QC06B | QC12B | QC30B | QC50B | QC100B |
|---|---|---|---|---|---|---|
| **Länge** | 0,45 m | 0,50 m | 1,2 m | 3,0 m | 5,0 m | 10,0 m |

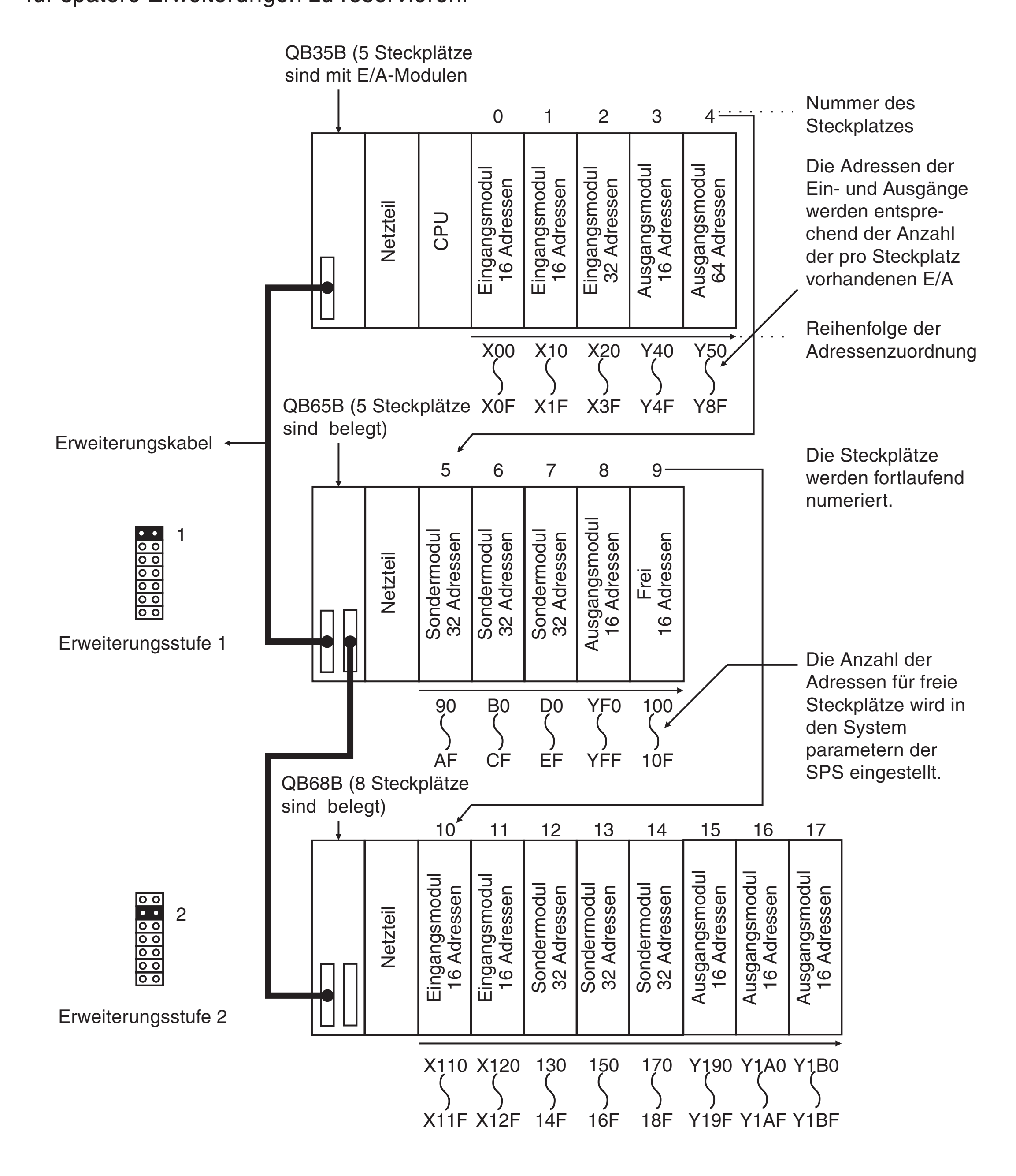
Zum Anschluss der Erweiterungsbaugruppenträger ohne eigenes Netzteil (Q52B, Q55B) wird das Kabel QC05B empfohlen.

### 3.2.2 Zuordnung der Adressen

Den Ein- und Ausgängen einer SPS müssen eindeutig gekennzeichnet werden, damit sie im Programm angesprochen werden können. Dazu wird jeder Ein- und Ausgang numeriert – es wird Ihnen eine Adresse zugewiesen (siehe auch Abschnitt 4.1). Die Adressen der Ein- und Ausgänge werden im hexadezimale Zahlensystem gezählt. (Mehr dazu später im Abschnitt 4.3.)

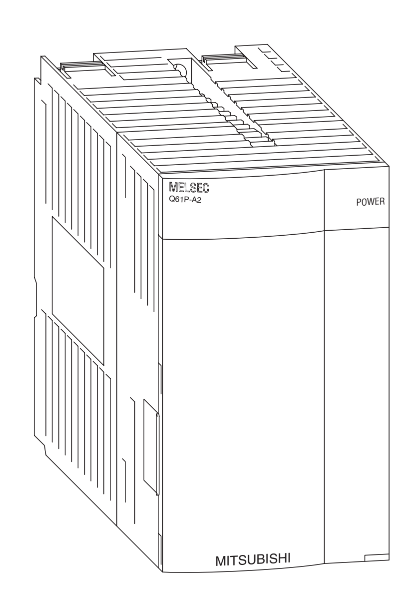
Eine CPU des MELSEC System Q erkennt automatisch die in den Haupt- und Erweiterungsbaugruppenträgern vorhandenen Steckplätze und ordnet die Adressen der Ein- und Ausgänge entsprechend zu.

Die Zuordnung kann jedoch auch mit Hilfe der Programmier-Software vom Anwender vorgenommen werden. Dadurch besteht die Möglichkeit, Steckplätze freizulassen oder Adressen für spätere Erweiterungen zu reservieren.




Die Erweiterungsstufe wird an den Erweiterungsbaugruppenträgern durch Steckbrücken festgelegt.

## 3.3 Netzteile




Das MELSEC System Q wird mit einer Gleichspannung von 5 Volt betrieben. Es stehen Netzteile mit Eingangsspannungen von 24 V DC oder 100 bis 240 V AC zur Verfügung.

Die Ausgangsspannung der Netzteile (5 V DC) wird direkt in den Baugruppenträger eingespeist und kann nicht an den Klemmen abgegriffen werden.

Beim Netzteil Q62P steht neben der Ausgangsspannung von 5 V noch ein zusätzlicher Gleichspannungsausgang von 24 V zur Verfügung, der beispielsweise zur Versorgung von Sensoren mit bis zu 0,6 A belastet werden kann.

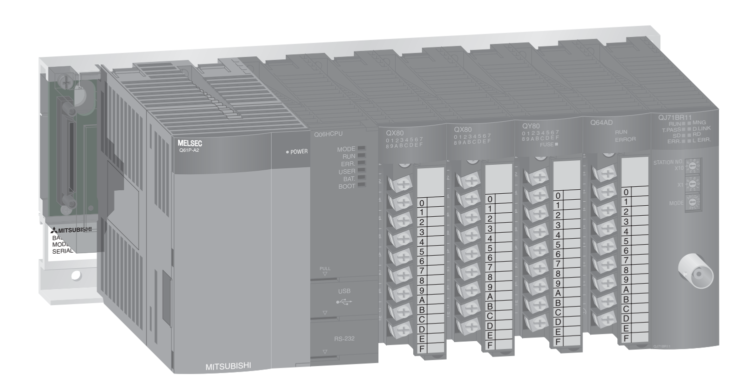
| Merkmal | Q63P | Q63RP | Q61P-A1 | Q61P-A2 | Q62P | | Q64P | Q64RP |
|---|---|---|---|---|---|---|---|---|
| Eingangs-spannung | 24 V DC | | 100–120 V AC | 200–220 V AC | 100–240 V AC | | 100–120 V AC<br>200–240 V AC | |
| Leistungs-aufnahme | 45 W | 65 W | 105 VA | 105 VA | 105 VA | | 105 VA | 160 VA |
| Ausgangs-spannung | 5 V DC | | 5 V DC | | 5 V DC | 24 V DC | 5 V DC | |
| Ausgangs-strom | 6 A | 8,5 A | 6 A | 6 A | 3 A | 0,6 A | 8,5 A | |

Die Netzteile Q63RP und Q64RP sind redundante Netzteile und können mit allen CPUs (ausser Q00JCPU) kombiniert werden. Für Netzteilredundanz werden 2 redundante Netzteile in einem redundanten Baugruppenträger benötigt. Dies erhöht die Verfügbarkeit des Systems, da bei Ausfall eines Netzteils die Spannungsversorgung vom zweiten Netzteil übernommen wird. Die redundanten Netzteile können während des Betriebs der SPS ohne Steuerungsunterbrechung ausgetauscht werden.

### Auswahl eines geeigneten Netzteils

Die Stromaufnahme der auf den Baugruppenträger installierten Module darf den Nennstrom, den das Netzteil liefern kann, nicht überschreiten. Falls dies der Fall ist, muss die Anzahl der Module im Baugruppenträger reduziert werden.

Beispiel zur Berechnung der Stromaufnahme:



| Modul | Art des Moduls | Stromaufnahme |
|---|---|---|
| Q06HCPU | CPU-Modul | 0,64 A |
| QX80 | Digital-Eingangsmodul | 0,16 A |
| QX80 | Digital-Eingangsmodul | 0,16 A |
| QY80 | Digital-Ausgangsmodul | 0,08 A |
| Q64AD | Analog-Eingangsmodul | 0,63 A |
| QJ71BR11 | MELSECNET/H-Modul | 0,75 A |
| Gesamtstromaufnahme | | 2,42 A |

Die Summe der Stromaufnahmen beträgt 2,42 A und liegt damit unter dem Nennstrom von 6 A, den das Netzteil liefern kann. Beim Betrieb der SPS werden daher keine Probleme auftreten.

## 3.4 Die CPU-Module

Das MELSEC System Q umfasst 19 verschiedene CPU-Module und bietet damit Leistung nach Maß. Bis zu vier CPU-Module können auf einen Baugruppenträger installiert werden und sich so die Steuerungs- und Kommunikationsaufgaben teilen. Wie bei den anderen Mitsubishi-Steuerungen wächst die Leistungsfähigkeit des MELSEC System Q mit der Applikation: Sie tauschen einfach die CPU oder fügen eine CPU hinzu.

Die CPU-Module werden unterschieden in:

- **SPS-CPUs**

  Eine SPS-CPU übernimmt im MELSEC System Q die „klassischen“ Aufgaben einer SPS. Sie führt das SPS-Programm aus, prüft den Zustand der Eingänge, steuert die Ausgänge und kommuniziert mit Sondermodulen.

- **Prozess-CPUs**

  Die Prozess-CPU-Module des MELSEC System Q haben die Funktionalität der SPS-CPUs und bieten zusätzlich erweiterte Regelungsfunktionen sowie 52 Prozessbefehlsfunktionen. Dadurch eignen sie sich für komplexe Aufgaben z. B. in der chemischen Industrie.

- **Redundante Prozess-CPUs**

  Neben allen Funktionen der Prozess-CPUs gewährleisten die redundanten CPUs des MELSEC System Q eine hohe Verfügbarkeit und Ausfallsicherheit.

  Eine redundante SPS besteht aus zwei Steuerungen mit identischer Konfiguration (Netzteil, CPU-Modul, Netzwerkmodule usw.), die mit einem Kabel verbunden sind. Eine SPS übernimmt die Steuerung, während die andere als Reservesystem in Bereitschaft steht. Bei einer Störung wird ohne Unterbrechung zwischen den beiden Steuerungen umgeschaltet und der Betrieb fortgesetzt. So werden Stillstandzeiten und Wiederanlaufkosten erheblich reduziert.

- **PC-CPU**

  Die PC-CPU ist ein vollwertiger kompakter Personal Computer, der auf dem Hauptbaugruppenträger installiert wird. Sie kann sowohl PC-typische Anwendungen als auch SPS-Anwendungen übernehmen. Er eignet sich somit als integrierter PC in der Steuerungstechnik z. B. für Visualisierung, Datenbanken, Log-Trace-Funktionen der Microsoft-Applikation oder zur Programmierung des System Q in einer Hochsprache. Darüberhinaus kann mit der optionalen Software SX-Controller das System als Soft-SPS nach IEC1131 gesteuert werden.

  Zur Verbindung mit der Peripherie können E/A- und Sondermodule aus dem MELSEC System Q verwendet werden.

- **C-CPU**

  Der C-Controller erlaubt die Integration und Programmierung der Automatisierungsplattform System Q unter C++. Mittels des Echtzeitbetriebssystems VxWorks, welches sich weltweit bewährt hat, können technologische Aufgaben leicht realisiert werden.

- **Motion-CPUs**

  Eine Motion-Controller-CPU steuert und synchronisiert die angeschlossenenen Servoverstärker und Servomotoren. In einem Motion-System muss immer mindestens eine SPS-CPU installiert sein. Erst durch die Kombination aus hochdynamischer Positioniersteuerung und SPS entsteht ein innovatives autarkes Bewegungssteuerungssystem.

  Während die Motion-CPU aufwändige Servo-Bewegungen steuert, bedient die SPS-CPU zeitgleich den maschinellen Betriebsablauf und die Kommunikation.

In diesem Einsteigerhandbuch wird nur auf die SPS-CPUs näher eingegangen. Informationen zu den anderen CPU-Modulen finden Sie im Technische Katalog MELSEC System Q, Art.-Nr. 136729 und in den Bedienungsanleitungen der einzelnen Module.

**SPS-CPUs**

- **Q00JCPU**

  CPU, ein Baugruppenträger mit fünf Steckplätzen und ein Netzteil bilden eine untrennbare, kompakte Einheit. Ein Multi-CPU-Betrieb ist bei der Q00JCPU nicht möglich.

  – Programmkapazität: 8 k Schritte
  – Ausführungszeit für eine logische Anweisung: 0,2 µs

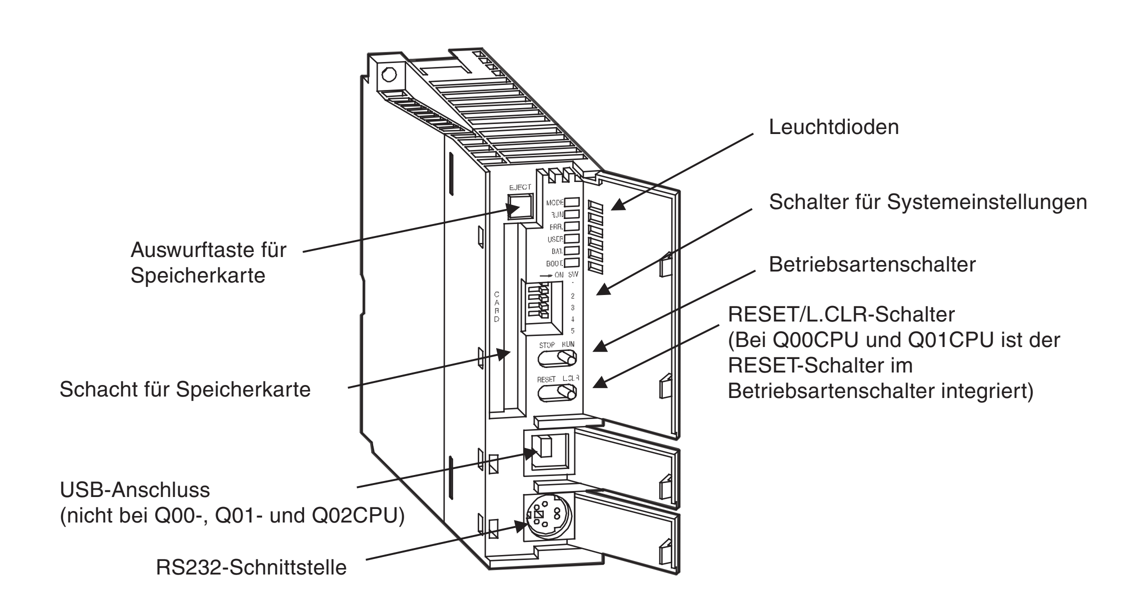
Alle folgenden SPS-CPUs sind für den Multi-CPU-Betrieb geeignet.

- **Q00CPU**
  – Programmkapazität: 8 k Schritte
  – Ausführungszeit für eine logische Anweisung: 0,16 µs
- **Q01CPU**
  – Programmkapazität: 14 k Schritte
  – Ausführungszeit für eine logische Anweisung: 0,10 µs
- **Q02CPU**
  – Programmkapazität: 28 k Schritte
  – Ausführungszeit für eine logische Anweisung: 0,079 µs
- **Q02HCPU**
  – Programmkapazität: 28 k Schritte (erweiterbar durch Speicherkarte)
  – Ausführungszeit für eine logische Anweisung: 0,034 µs
- **Q06HCPU**
  – Programmkapazität: 60 k Schritte (erweiterbar durch Speicherkarte)
  – Ausführungszeit für eine logische Anweisung: 0,034 µs
- **Q12HCPU**
  – Programmkapazität: 124 k Schritte (erweiterbar durch Speicherkarte)
  – Ausführungszeit für eine logische Anweisung: 0,034 µs
- **Q25HCPU**
  – Programmkapazität: 252 k Schritte (erweiterbar durch Speicherkarte)
  – Ausführungszeit für eine logische Anweisung: 0,034 µs

Die folgende Tabelle gibt Auskunft über die Erweiterungsmöglichkeiten und die Anzahl der Ein- und Ausgänge der SPS-CPUs.

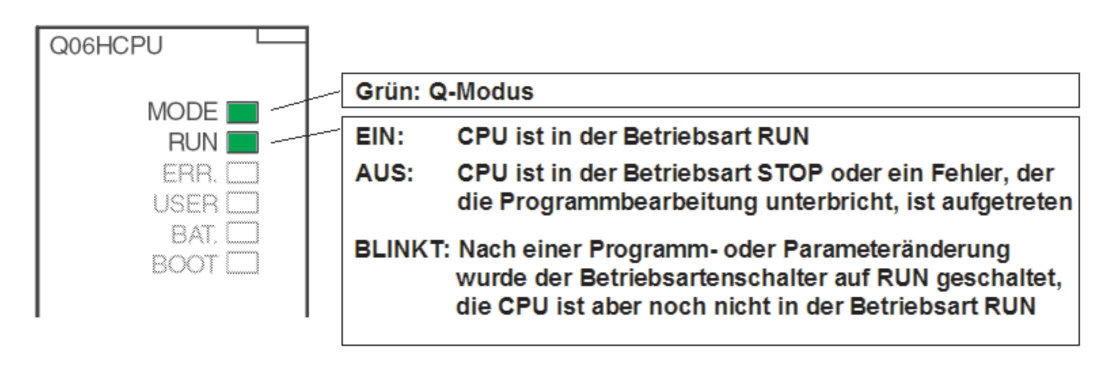
| CPU-Modul | Anzahl anschließbarer Erweiterungsbau-gruppenträger | Anzahl der installierbaren Module | Ein-/Ausgangsadressen | |
|---|---|---|---|---|
| | | | Zentral (In Haupt- und Erweiterungsbaugrup-penträgern) | Dezentral |
| Q00JCPU | 2 | 16 | 256 | 2048 |
| Q00CPU | 4 | 24 | 1024 | 2048 |
| Q01CPU | | | | |
| Q02CPU | 7 | 64 | 4096 | 8192 |
| Q02HCPU | | | | |
| Q06HCPU | | | | |
| Q12HCPU | | | | |
| Q25HCPU | | | | |

## 3.4.1 Bedienelemente der CPU-Module




### Leuchtdioden

– **MODE- und RUN-LED**



So schalten Sie die CPU nach einer Programm- oder Parameteränderung, die in der Betriebsart STOP ausgeführt wurde, in „RUN“:

① RESET/L.CLR-Schalter in Stellung „RESET“ schalten.

② RUN/STOP-Schalter in Stellung „RUN“ schalten.

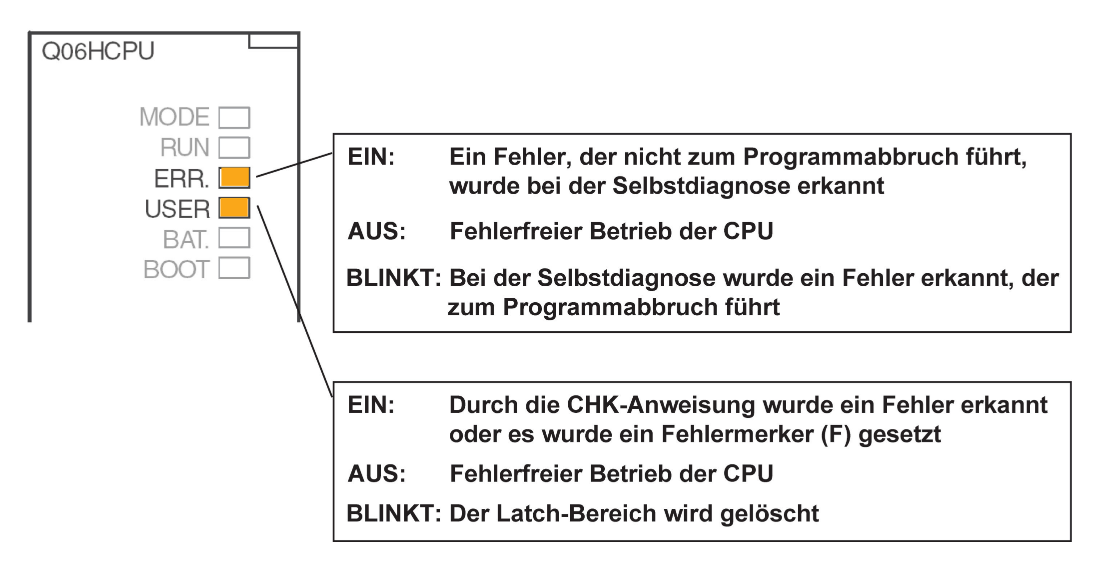
oder, falls kein Reset ausgeführt werden soll:

① RUN/STOP-Schalter von „STOP“ in Stellung „RUN“ schalten

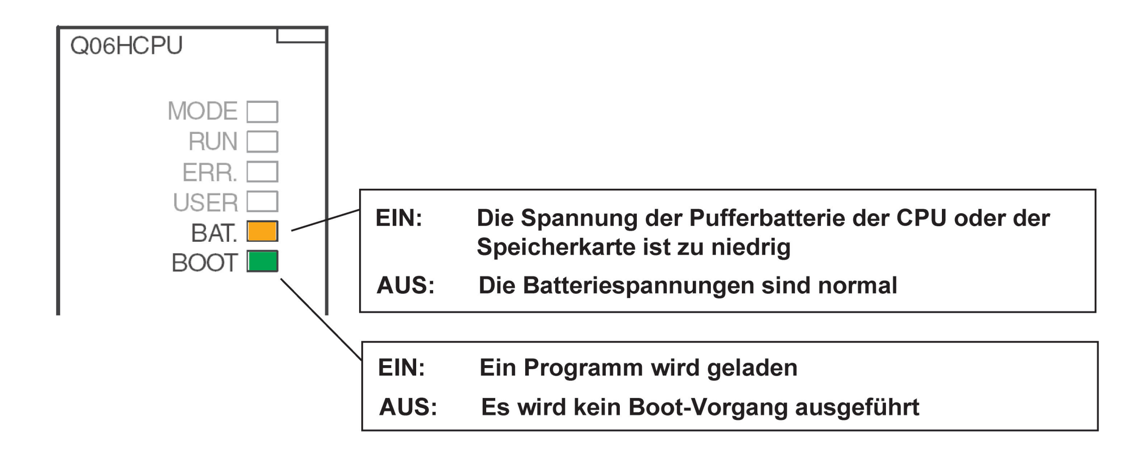
② RUN/STOP-Schalter dann wieder auf „STOP“ schalten

③ RUN/STOP-Schalter auf „RUN“ schalten.

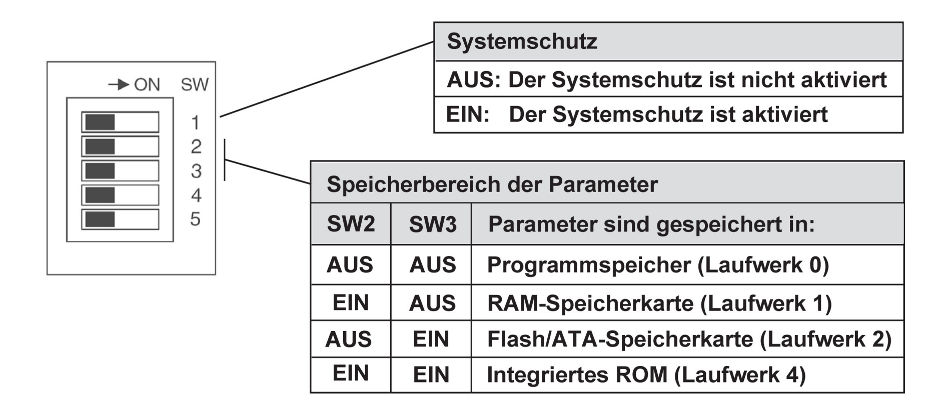
– **ERR.- und USER-LED**




– **BAT- und BOOT-LED**



### Systemschalter

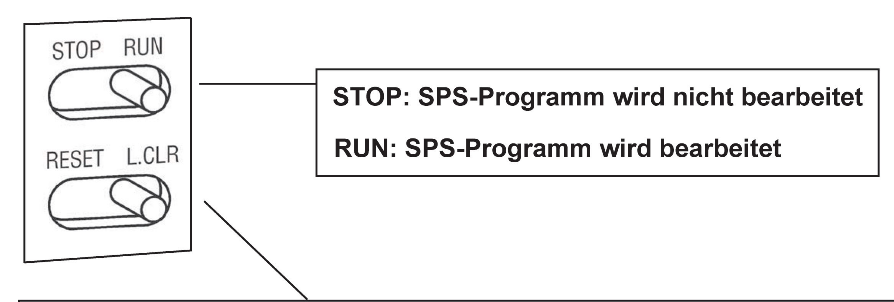



| SW2 | SW3 | Parameter sind gespeichert in: |
|---|---|---|
| AUS | AUS | Programmspeicher (Laufwerk 0) |
| EIN | AUS | RAM-Speicherkarte (Laufwerk 1) |
| AUS | EIN | Flash/ATA-Speicherkarte (Laufwerk 2) |
| EIN | EIN | Integriertes ROM (Laufwerk 4) |

Im integrierten RAM (Laufwerk 3) können keine Parameter gespeichert werden (siehe Abschnitt 3.4.2).

Bei der Auslieferung des CPU-Moduls sind alle Schalter in der Stellung „AUS“.

### RUN/STOP-Schalter, RESET/L.CLR-Schalter

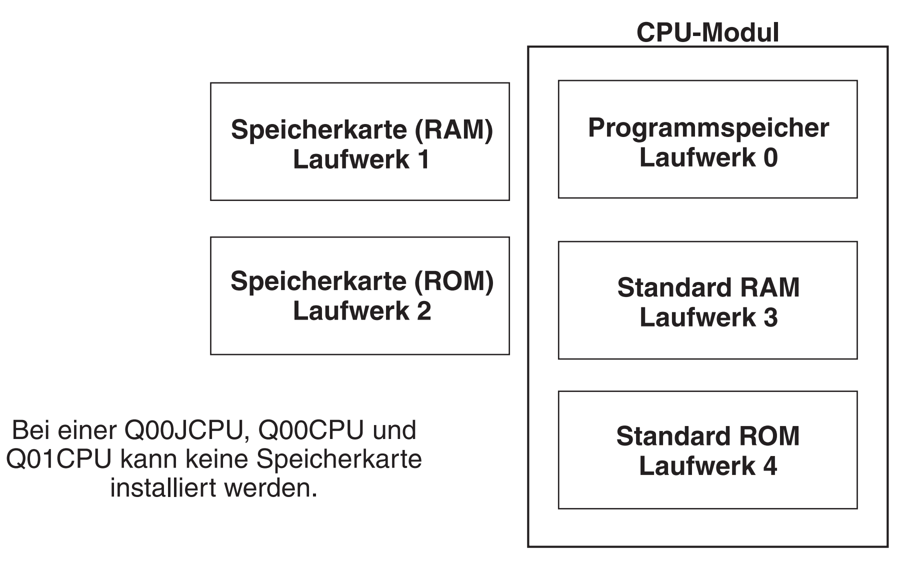



**RESET:** Zurücksetzen von Fehlermeldungen und Initialisierung der SPS
Nach einem Reset muss der Schalter wieder in die Mittelstellung gebracht werden.

**L.CLR:** Latch Clear, Operandendaten, die im parametriertem Latch-Bereich gespeichert sind, werden gelöscht (ausgeschaltet oder auf 0 gesetzt). (Nicht bei Q00CPU und Q01CPU)

### 3.4.2 Konfiguration des Speichers

Ein CPU des MELSEC System Q kann auf verschiedene interne Speicher zugreifen. Diese Speicher werden durch die Laufwerksnummer unterschieden. Zusätzlich sind die Hochleistungs-CPU-Module mit einem Steckplatz für eine Speicherkarte ausgestattet.




**Was kann wo gespeichert werden?**

- Q00JCPU, Q00CPU und Q01CPU

| Daten | Integrierter Speicher | | |
|---|---|---|---|
| | Programmspeicher (Laufwerk 0) | RAM (Laufwerk 3) | ROM (Laufwerk 4) |
| Programm | ● | ○ | ● |
| Parameter | ● | ○ | ● |
| Parameter für Sondermodule | ● | ○ | ● |
| Operandenkommentare | ● | ○ | ● |
| File-Register | ○ | ● | ○ |

● = Speicherung ist möglich

○ = Speicherung ist nicht möglich

● Q02CPU, Q02HCPU, Q06HCPU, Q12HCPU und Q25HCPU:

| Daten | Integrierter Speicher | | | Speicherkarten | | |
|---|---|---|---|---|---|---|
| | Programm-speicher (Laufwerk 0) | RAM (Laufwerk 3) | ROM (Laufwerk 4) | RAM (Laufwerk 1) | Flash ROM (Laufwerk 2) | ATA ROM (Laufwerk 2) |
| Programm | ● | ○ | ● | ● | ● | ● |
| Parameter | ● | ○ | ● | ● | ● | ● |
| Parameter für Sondermodule | ● | ○ | ● | ● | ● | ● |
| Operandenkommentare | ● | ○ | ● | ● | ● | ● |
| Initialisierungswerte | ● | ○ | ● | ● | ● | ● |
| File-Register | ○ | ● | ○ | ● | ● | ○ |
| Lokale Operanden | ○ | ● | ○ | ● | ○ | ○ |
| TRACE-Daten | ○ | ○ | ○ | ● | ○ | ○ |
| Fehlerhistorie | ○ | ○ | ○ | ● | ○ | ○ |
| Daten, die mit einer FWRITE-Anweisung eingetragen wurden | ○ | ○ | ○ | ○ | ○ | ● |

● = Speicherung ist möglich

○ = Speicherung ist nicht möglich

Ein Programm, das im Standard-ROM, in einer RAM-Speicherkarte oder einer ROM-Speicherkarte abgelegt ist, wird beim Einschalten der Versorgungsspannung in den Programmspeicher übertragen und ausgeführt. Für diesen Zweck muss der Programmspeicher über ausreichend freien Speicherplatz verfügen. Wenn Trace-Daten zur Beseitigung von Fehlern, Fehlerdaten oder Daten mit der FWRITE-Anweisung gespeichert werden sollen, muss eine Speicherkarte verwendet werden.

**Übersicht über die Daten, die gespeichert werden können**

– Programme
  Das Ablaufprogramm, das von der SPS-CPU ausgeführt wird. Wenn mehrere Programme gleichzeitig bearbeitet werden, werden diese auch im Programmspeicher eingetragen.

– Parameter
  Datei, die SPS- und Netzwerkparameter enthält.

– Parameter für Sondermodule
  Parameter, die mit einer Software zur Parametrierung der Sondermodule eingetragen wurden. Diese Datei existiert nur, wenn die entsprechende Software verwendet wurde.

– Operandenkommentare
  Datei mit Kommentaren, die den Operanden zugeordnet wurden

– Initialisierungswerte
  Datei mit Werten, die Operanden beim Einschalten der CPU annehmen sollen

– File-Register
  Datei mit File-Registern (R, ZR); Wenn unterschiedliche Dateinamen vergeben werden, können verschiedene File-Registerdateien gespeichert werden. File-Register können in einer ROM-Speicherkarte (Laufwerk 2), aber nicht in einer ATA-Speicherkarte (Q2MEM-8MBA/16MBA/32MBA) gespeichert werden. Aus einem File-Register in einem Flash-ROM kann nur gelesen werden, eine Änderung der Daten duch das Programm ist jedoch nicht möglich.

– Lokale Operanden
  Wenn mehrere Programme ausgeführt werden, werden lokale Operanden ausschließlich vom entsprechenden Programm benutzt. Bei Ausführung der Programme werden die lokalen Operanden aus dem Bereich der lokalen Daten in den Bereich der ausführbaren Operanden transferiert.

– TRACE-Daten
  Daten, die mit der TRACE-Funktion zur Fehlererkennung und –beseitigung erfasst worden sind.

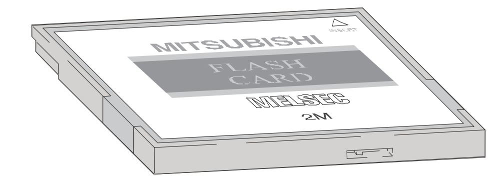
– Daten, die mit der FWRITE-Anweisung eingetragen wurden
  Diese Daten können nur auf einer ATA-Speicherkarte (Q2MEM-8MBA/16MBA/32MBA) abgelegt werden.

**Speicherkarten**

Mit Ausnahme der CPU-Module Q00JCPU, Q00CPU und Q01CPU können alle CPUs des MELSEC System Q mit Speicherkarten ausgestattet werden.

Der Speicherplatz steht nach der Formatierung der Speicherkarte zur Verfügung. Die Formatierung wird durch die Programmier-Software GX Developer oder GX IEC Developer vorgenommen.

Ein Programm, das in einer Speicherkarte abgelegt ist, wird beim Einschalten der Versorgungsspannung in den Programmspeicher übertragen und ausgeführt. In der Parametrierung kann das Anlaufverhalten (Boot file) eingestellt werden.

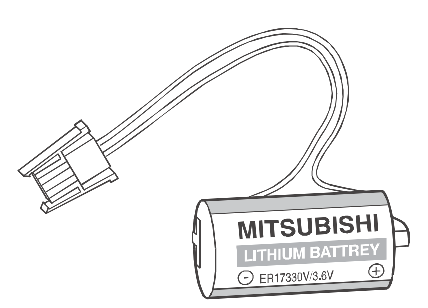



Die gespeicherten Daten können durch einen Schreibschutz gegen unbeabsichtigtes Löschen geschützt werden. In der SRAM-Speicherkarte puffert eine integrierte Batterie die gespeicherten Daten bei einem Spannungsausfall.

Die folgenden Speicherkarten sind erhältlich.

| Bezeichnung | Art des Speichers | Speicherkapazität [Byte] | Speicherkapazität [Dateien] | Anzahl der Schreibvorgänge |
|---|---|---|---|---|
| Q2MEM-1MBS | SRAM | 1011 k | 256 | Keine Beschränkung |
| Q2MEM-2MBS | | 2034 k | 288 | |
| Q2MEM-2MBF | Flash ROM | 2035 k | 288 | 100 000 |
| Q2MEM-4MBF | | 4079 k | | |
| Q2MEM-8MBA | ATA ROM | 7940 k | 512 | 1 000 000 |
| Q2MEM-16MBA | | 15932 k | | |
| Q2MEM-32MBA | | 31854 k | | |

### 3.4.3 Anschluss der Pufferbatterie des CPU-Moduls

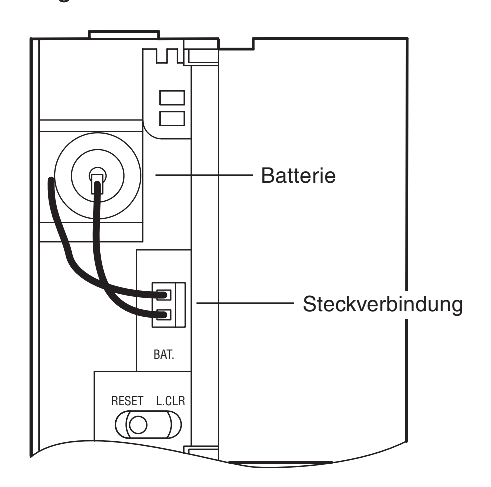



Die CPU-Module des MELSEC System Q sind mit einer Batterie ausgestattet. Bei einem Spannungsausfall kann sie den Programmspeicher, das integrierte RAM und die Uhr der CPU mehrere tausend Stunden (abhängig vom Typ der CPU) puffern.

Die Batterie sollte alle 10 Jahre gewechselt werden.

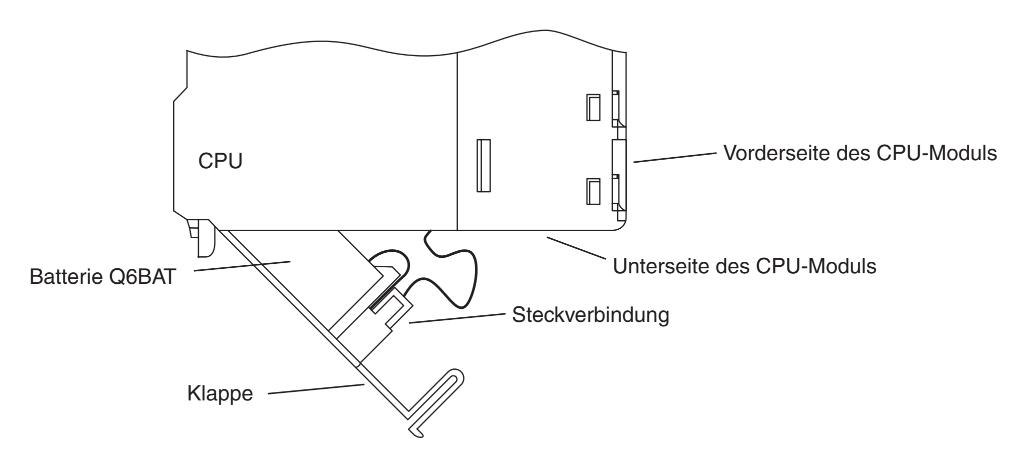
Die SRAM-Speicherkarten haben eine eigene Batterie vom Typ Q2MEM-BAT und sind dadurch unabhängig von der Batterie der CPU.

Bei der Auslieferung eines CPU-Moduls ist die Batterie zwar im CPU-Modul eingebaut, zum Schutz vor Kurzschlüssen und um eine Entladung zu vermeiden, ist die Steckverbindung zwischen Batterie und CPU aber getrennt. Vor der Inbetriebnahme der CPU muss die Batterie angeschlossen werden.




Bei einer Q00J-, Q00- oder Q01CPU erreichen Sie die Batterie nach dem Öffnen der oberen Klappe an der Vorderseite des CPU-Moduls.

Bei den anderen CPU-Typen erfolgt die Montage der Batterie von der Unterseite des CPU-Moduls aus.




Zum Anschluss der Batterie öffnen Sie das Batteriefach der CPU und vergewissern Sie sich, dass die Batterie korrekt eingesetzt ist. Verbinden Sie anschließend den Stecker der Batterie mit dem Gegenstück in der Batteriehalterung bzw. im CPU-Modul. Prüfen Sie bei der Q02(H)-, Q06H-, Q12(P)H- und Q25(P)HCPU, dass die Steckverbindung in die dafür vorgesehene Halterung im Batteriefach eingesetzt ist.

## 3.5 Digitale Ein- und Ausgangsmodule

Ein- und Ausgangsmodule verbinden die CPU einer SPS mit dem zu steuernden Prozess. Während digitale Eingangsmodule die Signale externer Geräte in eine EIN/AUS-Information für die CPU umwandeln, können externe Schaltglieder durch digitale Ausgangsmodule ein- oder ausgeschaltet werden.

**Eingangssignale** können von einer Vielzahl von Sensoren oder Geräten stammen:

- Drucktaster
- Drehschalter mit mehreren Stellungen
- Schlüsselschalter
- Endschalter
- Niveauschalter
- Sensoren zur Durchflussüberwachung
- Lichtschranken oder Lichttaster
- Näherungsschalter (Induktiv oder kapazitiv), Näherungsschalter sind in der Regel mit einem Transistorausgangs ausgestattet, der entweder plus- oder minusschaltend ausgelegt ist.

Mit **Ausgangssignalen** werden zum Beispiel angesteuert:

- Schütze
- Meldeleuchten
- Magnetventile
- Eingänge externer Geräte wie z. B. Frequenzumrichter

### Übersicht der digitalen E/A-Module

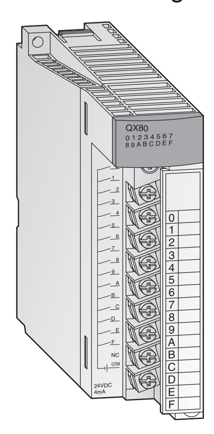
| Modultyp | | Anzahl der Ein- und Ausgänge | | | |
|---|---|---|---|---|---|
| | | 8 | 16 | 32 | 64 |
| Eingangsmodule | 120 V AC | ○ | ● | ○ | ○ |
| | 240 V AC | ● | ○ | ○ | ○ |
| | 48 V AC/DC | ○ | ● | ○ | ○ |
| | 24 V DC | ○ | ● | ● | ● |
| | 24 V DC (schnelle Eingänge) | ● | ○ | ○ | ○ |
| | 5 V DC / 12 V DC | ○ | ● | ● | ● |
| Ausgangsmodule | Relais | ● | ● | ○ | ○ |
| | Relais mit getrennten Kontakten | ● | ○ | ○ | ○ |
| | Triac-Ausgänge | ○ | ● | ○ | ○ |
| | Transistor-Ausgänge (minusschaltend) | ● | ● | ● | ● |
| | Transistor-Ausgänge (plusschaltend) | ○ | ● | ● | ○ |
| Kombinierte Ein-/Ausgangsmodule | | ● | ○ | ● | ○ |

● = Es ist ein Modul verfügbar

○ = Es steht kein Modul zur Verfügung

## 3.5.1 Digital-Eingangsmodule

Es stehen Digital-Eingangsmodule für verschiedene Eingangsspannungen zur Verfügung:




| Eingänge / Eingangsspannung | Eingangsmodul im MELSEC System Q | | | |
|---|---|---|---|---|
| | 8 | 16 | 32 | 64 |
| 5 – 12 V DC | | QX70 | QX71 | QX72 |
| 24 V DC | | QX40<br>QX80 | QX41<br>QX81 | QX42<br>QX82 |
| 24 V DC<br>(Interrupt-Modul) | | QI60 | | |
| 48 V AC/DC | | QX50 | | |
| 100 – 120 V AC | | QX10 | | |
| 100 – 240 V AC | QX28 | | | |

Bei den Eingangs-Modulen mit 8 oder 16 Eingängen erfolgt der Anschluss der externen Signale über abnehmbare Klemmleisten mit Schraubklemmen. Module mit 32 oder 64 Eingängen werden über Stecker angeschlossen.

### Grundsätzliches über digitale Eingangsmodule

Alle Eingänge sind durch Optokoppler isoliert. Dadurch wird die empfindliche Elektronik der SPS nicht durch elektromagnetische Störungen beeinflusst, die durch externe Geräte verursacht werden.

Ein anderes oft anzutreffendes Problem ist das Prellen der Kontakte von mechanischen Schaltern. Damit sich diese Störungen nicht auf die SPS auswirken, werden die Eingangssignale gefiltert. Ein geänderter Signalzustand wird nur erfasst, wenn er für eine bestimmte Zeit am Eingang anliegt. Kurzzeitige Störsignale werden dadurch von der SPS nicht als Eingangssignale interpretiert.

**HINWEIS** Für Standard-Eingangsmodule ist die Filterzeit auf 10 ms voreingestellt. Diese Voreinstellung kann aber für jedes installierte Modul in den Parametern im Bereich von 1 ms bis 70 ms geändert werden. Bitte beachten Sie hierzu die technischen Daten der Module.

Die eingestellte Filterzeit beeinflusst auch die Reaktionszeit der SPS und sollte daher bei der Programmierung berücksichtigt werden. Bei einer kurzen Filterzeit wird zwar die Reaktionszeit der SPS reduziert, gleichzeitig steigt aber die Empfindlichkeit gegenüber externen Störeinflüssen. Die Eingangsignale sollten in diesem Fall über abgeschirmte Leitungen geführt und diese Signalleitungen sollten getrennt von Leitungen verlegt werden, die potentielle Störquellen darstellen. Falls sehr kurze Reaktionszeiten gefordert sind, sollten spezielle Module wie das Interrupt-Modul QI60 eingesetzt werden.

Damit die SPS einen eingeschalteten Eingang erkennt, muss ein Mindeststrom in diesen Eingang (oder aus dem Eingang heraus) fließen. Dieser Strom hängt vom Typ des Eingangsmoduls ab und beträgt in den meisten Fällen 3 mA. Wird – auch bei vermeintlich eingeschaltetem Eingang – dieser Strom nicht erreicht, bleibt der Eingang für die CPU weiterhin ausgeschaltet. Der Eingangsstrom wird durch den Innenwiderstand des Eingangsmoduls begrenzt. Fließt durch eine zu hohe Eingangsspannung ein zu großer Eingangsstrom, wird das Eingangsmodul beschädigt. Eingangsströme von bis zu 7 mA sind zulässig.

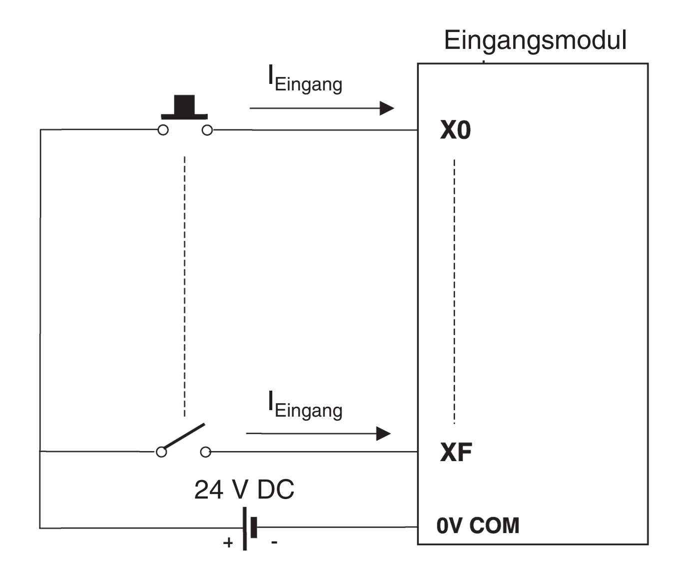
Die SPS-CPU erfasst den Zustand der Eingänge am Anfang der zyklischen Programmbearbeitung und speichert sie. Im Programm werden nur die gespeicherten Zustände verarbeitet. Erst vor der erneuten Bearbeitung des Programms werden die Eingangszustände wieder aktualisiert.

### Plus- und minusschaltende Eingänge

Im MELSEC System Q stehen Gleichspannungs-Eingangsmodule für plus- oder minusschaltende Sensoren zur Verfügung. An einigen Modulen wie z. B. dem QX71 können aber auch wahlweise plus- oder minusschaltende Sensoren angeschlossen werden. Im angelsächsischem Sprachraum wird bei plus- und minusschaltenden Gebern von „Source“ (Stromquelle) bzw. „Sink“ (Stromsenke) gesprochen. Diese Bezeichnungen beziehen sich auf die Richtung, in die der Strom bei eingeschaltetem Eingang fließt.

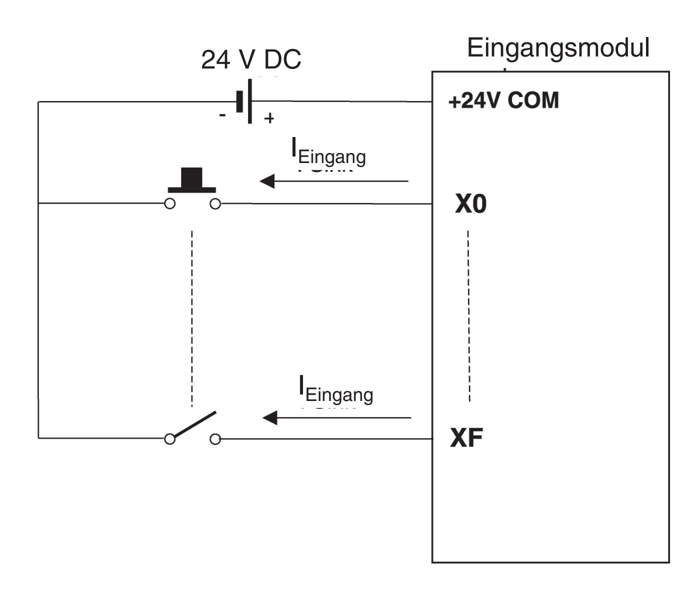
#### Anschluss plusschaltender Geber („Source“)

Ein plusschaltender Geber verbindet den Pluspol einer Spannungsquelle mit einem SPS-Eingang. Der Minuspol der Spannungsquelle bildet das gemeinsame Bezugspotential aller Eingänge einer Gruppe. Bei eingeschaltetem Geber fließt ein Strom in das Eingangsmodul, daher die englische Bezeichnung „Source“ – der Geber arbeitet als Stromquelle.




#### Anschluss minusschaltender Geber („Sink“)

Ein minusschaltender Geber verbindet den Minuspol einer Spannungsquelle mit einem SPS-Eingang. Das gemeinsame Bezugspotential aller Eingänge einer Gruppe ist der Pluspol der Spannungsquelle. Bei eingeschaltetem Geber fließt ein Strom aus dem Eingangsmodul, der Geber wirkt als Stromsenke, daher die englische Bezeichnung „Sink“.



### Näherungsschalter und optische Sensoren

**Näherungsschalter** sind berührungslose Schalter. Sie geben ein Signal an die SPS, wenn sich ein Objekt dem Schalter mit einem geringen Abstand nähert. Der Schalter muss vom zu erfassenden Objekt nicht berührt werden. Dadurch ergeben sich viele Einsatzmöglichkeiten in der Anlagenautomatisierung. Näherungsschalter können induktiv oder kapazitiv arbeiten.

Weit verbreitet in industriellen Steuerungen sind auch **optische Sensoren** in Form von Lichtschranken oder Lichttaster. (Lichtschranken benötigen einen Spiegel, der den Lichtstrahl reflektiert. Bei Lichttastern wird das ausgesendete Licht vom Objekt zurückgestrahlt.)

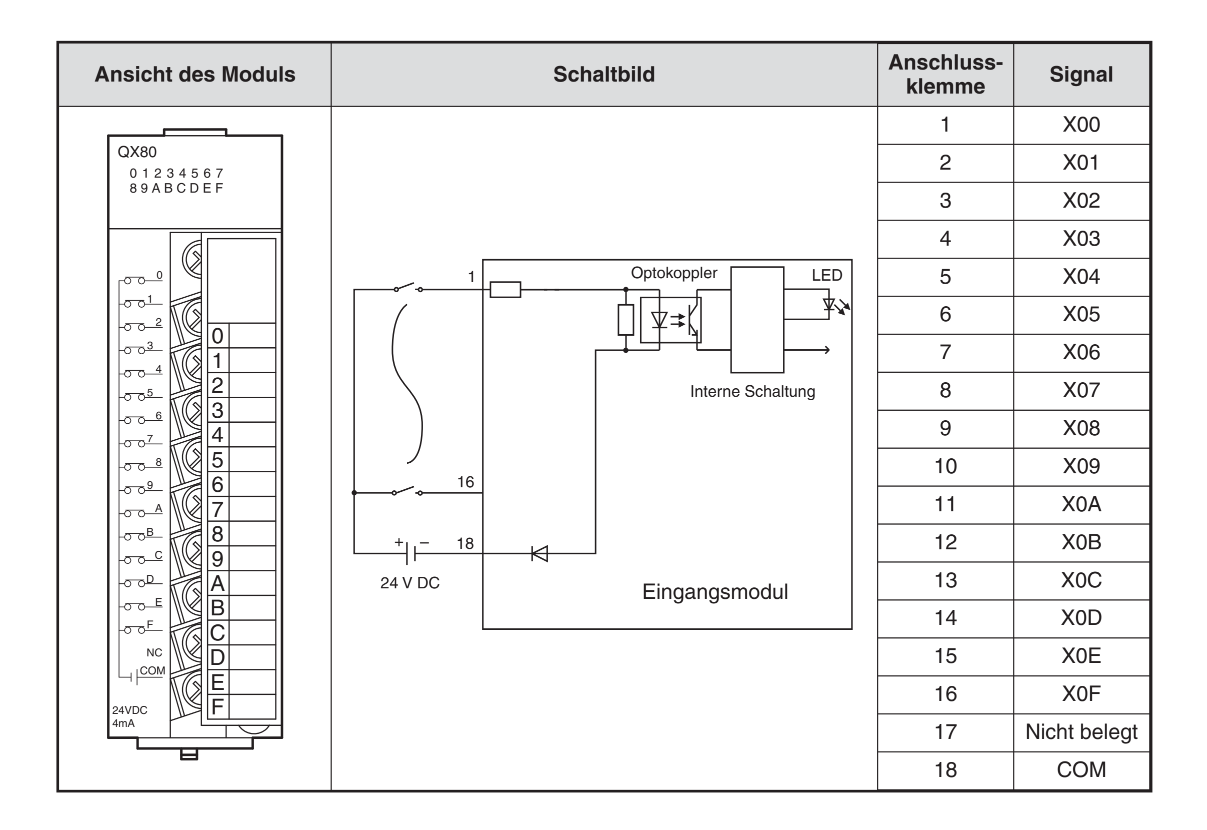
Näherungsschalter und Lichtschranken oder -taster sind mit einer internen Elektronik ausgestattet, die in den meisten Fällen eine Versorgungsspannung von 24 V DC benötigt. Die Ausgänge dieser elektronischen Schalter sind in der Regel als Transistorausgänge ausgelegt und schalten entweder plus oder minus:

- PNP-Transistorausgang: plusschaltend (Source)
- NPN-Transistorausgang: minusschaltend (Sink)

### Beispiel für ein Eingangsmodul für plusschaltende Geber

| Merkmal | | Technische Daten |
|---|---|---|
| Bezeichnung des Moduls | | QX80 |
| Eingänge | | 16 |
| Isolation | | durch Optokoppler |
| Eingangsnennspannung | | 24 V DC (+20/-15%, Welligkeit bis 5%) |
| Eingangsstrom | | ca. 4 mA |
| Gleichzeitig schaltbare Eingänge | | 100 % (Alle Eingänge können gleichzeitig eingeschaltet sein.) |
| Einschaltstromspitze | | Max. 200 mA für 1 ms (bei 132 V AC) |
| Spannung und -strom für EIN | | ≥ 19 V DC / ≥ 3 mA |
| Spannung und -strom für AUS | | ≤ 11 V DC / ≤ 1,7 mA |
| Eingangswiderstand | | ca. 5,6 kΩ |
| Ansprechzeit | AUS → EIN | 1, 5, 10, 20, 70 ms (parametrierbar, Voreinstellung: 10 ms)* |
| | EIN → AUS | |
| Durchschlagfestigkeit | | 560 V AC Effektivwert für 3 Zyklen (Einsatzhöhe 2000 m) |
| Isolationswiderstand | | ≥ 10 MΩ (Messung mit Isolationsprüfgerät) |
| Störfestigkeit | | Geprüft mit Störsimulator (Spitzenwert der Störspannung: 500 V, Einschaltdauer der Störspannung: 1μs, Frequenz der Störspannung: 25 bis 60 Hz) |
| | | Hochfrequente, nicht periodische Störspannung (IEC61000-4-4): 1kV |
| Eingangsgruppen | | 1 Gruppe mit 16 Eingängen, Bezugspotential: Anschlussklemme 18 |
| Statusanzeige der Eingänge | | Eine LED pro Eingang |
| Anschluss der Verdrahtung | | Klemmblock mit 18 Schraubklemmen (M3 x 6) |
| Empfohlener Leitungsquerschnitt | | 0,3 bis 0,75 $mm^2$, Drahtdurchmesser: 2,8 mm |
| Interne Stromaufnahme (5 V DC) | | 50 mA (wenn alle Eingänge eingeschaltet sind) |
| Gewicht | | 0,16 kg |

* Die Ansprechzeiten von AUS nach EIN und von EIN nach AUS können nicht separat eingestellt werden.




| Anschluss-klemme | Signal |
|---|---|
| 1 | X00 |
| 2 | X01 |
| 3 | X02 |
| 4 | X03 |
| 5 | X04 |
| 6 | X05 |
| 7 | X06 |
| 8 | X07 |
| 9 | X08 |
| 10 | X09 |
| 11 | X0A |
| 12 | X0B |
| 13 | X0C |
| 14 | X0D |
| 15 | X0E |
| 16 | X0F |
| 17 | Nicht belegt |
| 18 | COM |

### Funktion eines Eingangsmoduls mit plusschaltenden Gebern

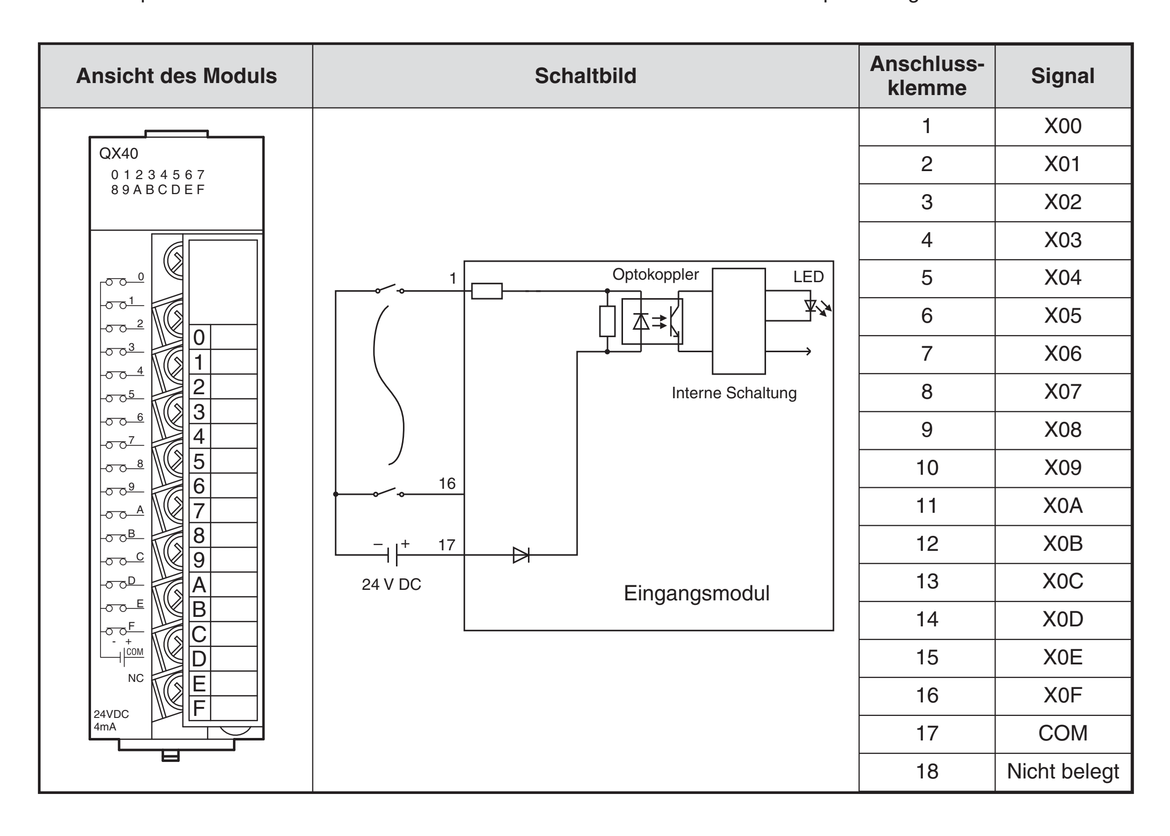
Wird ein an einem Eingangsmodul QX80 (siehe vorherige Seite) angeschlossener Geber, wie z. B. ein Drucktaster mit Schließerfunktion, betätigt, wird der SPS-Eingang eingeschaltet. Dabei laufen die folgenden Vorgänge ab, die sich auf das Schaltbild auf der vorherigen Seite beziehen:

- Bei betätigtem Taster wird der Pluspol der externen 24-Volt-Spannungsquelle mit Anschluss 1 des Eingangsmoduls verbunden.
- Anschluss 1 ist über einen Widerstand und die Leuchtdiode des Optokopplers mit dem Minuspol der externen Spannungsquelle (Anschluss 18) verbunden. Dadurch fließt ein Strom durch die LED des Optokopplers.
- Der fließende Strom bringt die LED zum Leuchten. Dadurch wird der Fototransistor des Optokopplers durchgesteuert.
- Durch den Optokoppler wird die externe Eingangsspannung von der Versorgungsspannung der SPS getrennt. Dadurch werden Störungen, die in industriellen Umgebungen oft diese externen Gleichspannungen überlagern, nicht auf die Versorgungsspannung der SPS übertragen. Durch den Optokoppler wird der Eingang zudem unempfindlicher gegen Störungen.
- Wenn der Fototransistor des Optokopplers durchsteuert, wird an die Eingangslogik des Moduls ein Signal übermittelt. In diesem Beispiel registriert die Elektronik, dass der Eingang X0 eingeschaltet ist. Die Leuchtdiode an der Vorderseite des Eingangsmoduls leuchtet in diesem Fall auf und signalisiert diesen Signalzustand.

## Beispiel für ein Eingangsmodul für minusschaltende Geber

| Merkmal | | Technische Daten |
|---|---|---|
| Bezeichnung des Moduls | | QX40 |
| Eingänge | | 16 |
| Isolation | | durch Optokoppler |
| Eingangsnennspannung | | 24 V DC (+20/-15%, Welligkeit bis 5%) |
| Eingangsstrom | | ca. 4 mA |
| Gleichzeitig schaltbare Eingänge | | 100 % (Alle Eingänge können gleichzeitig eingeschaltet sein.) |
| Einschaltstromspitze | | Max. 200 mA für 1 ms (bei 132 V AC) |
| Spannung und -strom für EIN | | ≥ 19 V DC / ≥ 3 mA |
| Spannung und -strom für AUS | | ≤ 11 V DC / ≤ 1,7 mA |
| Eingangswiderstand | | ca. 5,6 kΩ |
| Ansprechzeit | AUS → EIN | 1, 5, 10, 20, 70 ms (parametrierbar, Voreinstellung: 10 ms)* |
| | EIN → AUS | |
| Durchschlagfestigkeit | | 560 V AC Effektivwert für 3 Zyklen (Einsatzhöhe 2000 m) |
| Isolationswiderstand | | ≥ 10 MΩ (Messung mit Isolationsprüfgerät) |
| Störfestigkeit | | Geprüft mit Störsimulator (Spitzenwert der Störspannung: 500 V, Einschaltdauer der Störspannung: 1 $\mu$s, Frequenz der Störspannung: 25 bis 60 Hz) |
| | | Hochfrequente, nicht periodische Störspannung (IEC61000-4-4): 1kV |
| Eingangsgruppen | | 1 Gruppe mit 16 Eingängen, Bezugspotential: Anschlussklemme 17 |
| Statusanzeige der Eingänge | | Eine LED pro Eingang |
| Anschluss der Verdrahtung | | Klemmblock mit 18 Schraubklemmen (M3 x 6) |
| Empfohlener Leitungsquerschnitt | | 0,3 bis 0,75 $mm^2$, Drahtdurchmesser: 2,8 mm |
| Interne Stromaufnahme (5 V DC) | | 50 mA (wenn alle Eingänge eingeschaltet sind) |
| Gewicht | | 0,16 kg |

* Die Ansprechzeiten von AUS nach EIN und von EIN nach AUS können nicht separat eingestellt werden.




| Ansicht des Moduls | Schaltbild | Anschluss-klemme | Signal |
|---|---|---|---|
| | | 1 | X00 |
| | | 2 | X01 |
| | | 3 | X02 |
| | | 4 | X03 |
| | | 5 | X04 |
| | | 6 | X05 |
| | | 7 | X06 |
| | | 8 | X07 |
| | | 9 | X08 |
| | | 10 | X09 |
| | | 11 | X0A |
| | | 12 | X0B |
| | | 13 | X0C |
| | | 14 | X0D |
| | | 15 | X0E |
| | | 16 | X0F |
| | | 17 | COM |
| | | 18 | Nicht belegt |

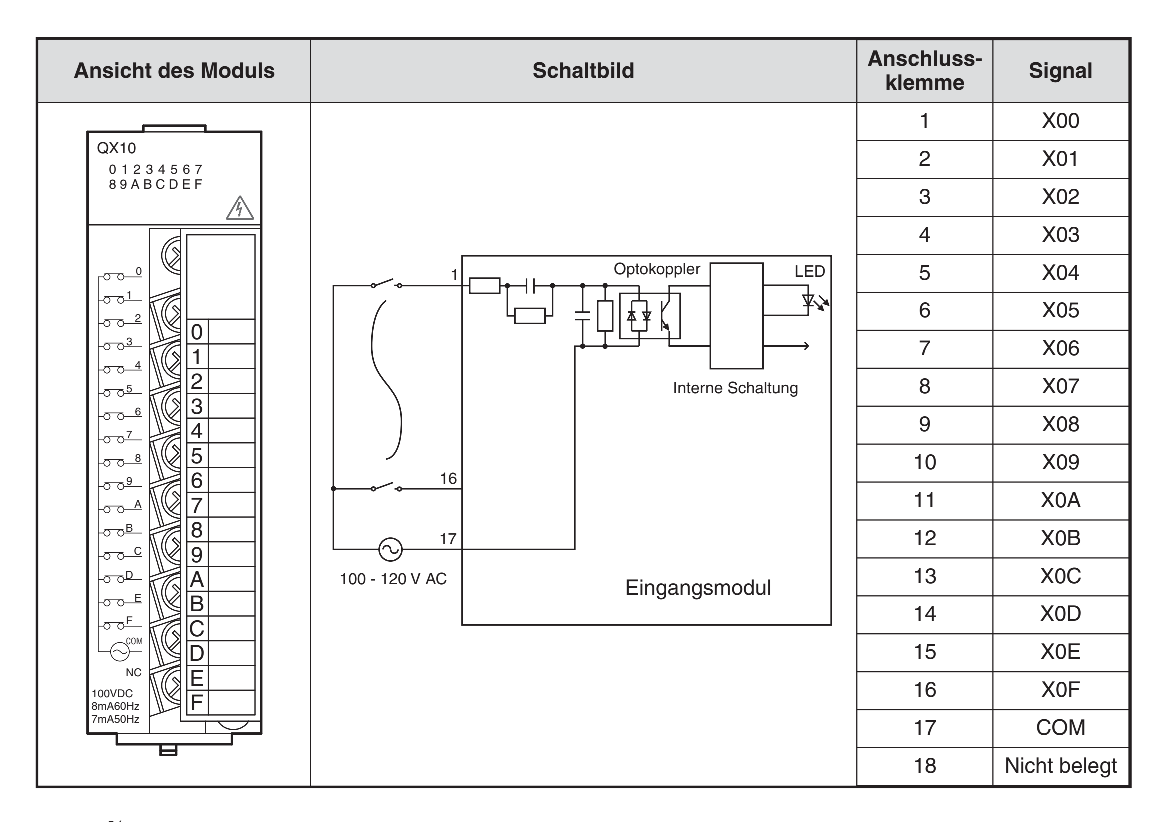
### Funktion eines Eingangsmoduls mit minusschaltenden Gebern

Wenn der im Schaltbild auf der vorherigen Seite an der Klemme 1 angeschlossene Schalter betätigt wird, fließt der Strom wie folgt:

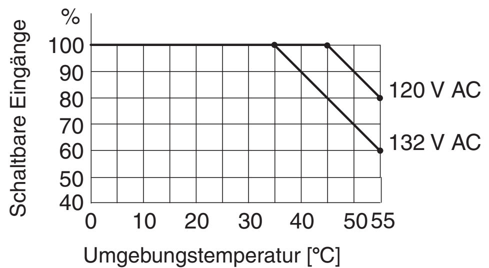
- Aus dem Pluspol der externen 24-Volt-Spannungsquelle in den Anschluss für das Bezugspotential (Klemme 17).
- Durch die Leuchtdiode des Optokopplers und den Vorwiderstand zur Klemme 1 (Anschluss für Eingangs X0) des Eingangsmoduls.
- Der Strom durch die LED des Optokopplers bringt diese zum Leuchten. Dadurch wird der Fototransistor des Optokopplers eingeschaltet.
- Wenn der Fototransistor des Optokopplers durchsteuert, wird an die Eingangslogik des Moduls ein Signal übermittelt. In diesem Beispiel registriert die Elektronik, dass der Eingang X0 eingeschaltet ist. Die Leuchtdiode an der Vorderseite des Eingangsmoduls leuchtet in diesem Fall auf und signalisiert diesen Signalzustand.
- Aus dem Anschluss für X0 fließt der Strom über den betätigtem Schalter zum Minuspol der externen Spannungsquelle.

### Beispiel für ein Wechselspannungs-Eingangsmodul

| Merkmal | | Technische Daten |
|---|---|---|
| Bezeichnung des Moduls | | QX10 |
| Eingänge | | 16 |
| Isolation | | Durch Optokoppler |
| Eingangsnennspannung | | 100 — 120 V AC (+10/-15 %) 50/60 Hz (±3Hz) (Verzerrungen bis 5 %) |
| Eingangsstrom | | ca. 8 mA bei 100 V AC, 60 Hz; ca. 7 mA bei 100 V AC, 50 Hz |
| Gleichzeitig schaltbare Eingänge | | siehe Diagramm |
| Einschaltstromspitze | | Max. 200 mA für 1 ms (bei 132 V AC) |
| Spannung und -strom für EIN | | ≥ 80 V AC / ≥ 5 mA (50 Hz, 60 Hz) |
| Spannung und -strom für AUS | | ≤ 30 V AC / ≤ 1 mA (50 Hz, 60 Hz) |
| Eingangswiderstand | | ca. 15 kΩ bei 60 Hz, ca. 18 kΩ bei 50 Hz |
| Ansprechzeit | AUS → EIN | ≤ 15 ms (100 V AC, 50 Hz, 60 Hz) |
| | EIN → AUS | ≤ 20 ms (100 V AC, 50 Hz, 60 Hz) |
| Durchschlagfestigkeit | | 1780 V AC Effektivwert für 3 Zyklen (Einsatzhöhe 2000 m) |
| Isolationswiderstand | | ≥ 10 MΩ (Messung mit Isolationsprüfgerät) |
| Störfestigkeit | | Geprüft mit Störsimulator (Spitzenwert der Störspannung: 1500 V, Einschaltdauer der Störspannung: 1μs, Frequenz der Störspannung: 25 bis 60 Hz) |
| | | Hochfrequente, nicht periodische Störspannung (IEC61000-4-4): 1kV |
| Eingangsgruppen | | 1 Gruppe mit 16 Eingängen, Bezugspotential: Anschlussklemme 17 |
| Statusanzeige der Eingänge | | Eine LED pro Eingang |
| Anschluss der Verdrahtung | | Klemmblock mit 18 Schraubklemmen (M3 x 6) |
| Empfohlener Leitungsquerschnitt | | 0,3 bis 0,75 $mm^2$, Drahtdurchmesser: 2,8 mm |
| Interne Stromaufnahme (5 V DC) | | 50 mA |
| Gewicht | | 0,17 kg |




| Ansicht des Moduls | Schaltbild | Anschluss-klemme | Signal |
|---|---|---|---|
| | | 1 | X00 |
| | | 2 | X01 |
| | | 3 | X02 |
| | | 4 | X03 |
| | | 5 | X04 |
| | | 6 | X05 |
| | | 7 | X06 |
| | | 8 | X07 |
| | | 9 | X08 |
| | | 10 | X09 |
| | | 11 | X0A |
| | | 12 | X0B |
| | | 13 | X0C |
| | | 14 | X0D |
| | | 15 | X0E |
| | | 16 | X0F |
| | | 17 | COM |
| | | 18 | Nicht belegt |

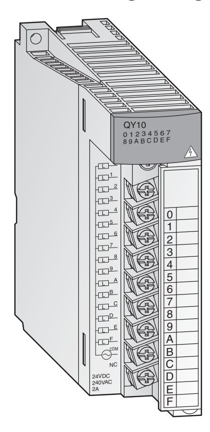



Die Anzahl der gleichzeitig schaltbare Eingänge hängt beim Modul QX10 von der Umgebungstemperatur ab.

Bei Eingangsmodulen für Wechselspannungen sollte dieselbe Spannung , die auch die SPS versorgt (z. B. 230 V AC), zum Schalten der Eingänge verwendet werden. Dadurch wird verhindert, das an den Eingängen eine falsche Spannung angeschlossen wird.

## 3.5.2 Digital-Ausgangsmodule

Durch unterschiedliche Schaltelemente bieten die Ausgangsmodule eine Lösung für jede Steuerungsaufgabe.



| Ausgangstyp | Ausgänge / Nennspannung | Ausgangsmodul | | | |
|---|---|---|---|---|---|
| | | 8 | 16 | 32 | 64 |
| Relais | 24 V DC / 240 V AC | QY18A | QY10 | | |
| Triac | 100 – 240 V AC | | QY22 | | |
| Transistor | 5 / 12 V DC | | QY70 | QY71 | |
| | 12 / 24 V DC | | QY80 | QY81P | QY42P |
| | 5 – 24 V DC | QY68A | | | |

Module mit 8 oder 16 Ausgängen besitzen abnehmbare Klemmleisten mit Schraubklemmen zum Anschluss der Ausgangssignale. Module mit 32 Ausgängen werden über Stecker angeschlossen.

### Ausgangstypen

Die digitalen Ausgangsmodule des MELSEC System Q sind in vier verschiedenen Ausgangstypen erhältlich.

- Relais
- Triac
- Transistor (plusschaltend)
- Transistor (minusschaltend)

| Typ | Vorteile | Nachteile |
|---|---|---|
| Relais | ● Ein Modul kann verschieden Spannungen schalten.<br>● Potentialfreie Kontakte<br>● Schalten hoher Ströme ist möglich | ● Langsam (max. 1 Hz)<br>● Begrenzte Lebendauer (elektromechanisch)<br>● Gefahr von verbrannten Schaltkontakten<br>● Laut (das Schalten ist hörbar) |
| Triac | ● Zuverlässig<br>● Hohe Schaltgeschwindigkeit<br>● Geeignet für hohe Anforderungen | ● Schaltet nur Wechselspannung<br>● max. Schaltstrom 0,6 A pro Ausgang<br>● Benötigt 10 ms Schaltzeit bei 50 Hz AC |
| Transistor | ● Sehr zuverlässig<br>● Sehr hohe Schaltgeschwindigkeit<br>● Besonders geeignet für hohe Anforderungen | ● Schaltet nur niedrige Gleichspannungen<br>● max. Schaltstrom 0,1 A pro Ausgang |

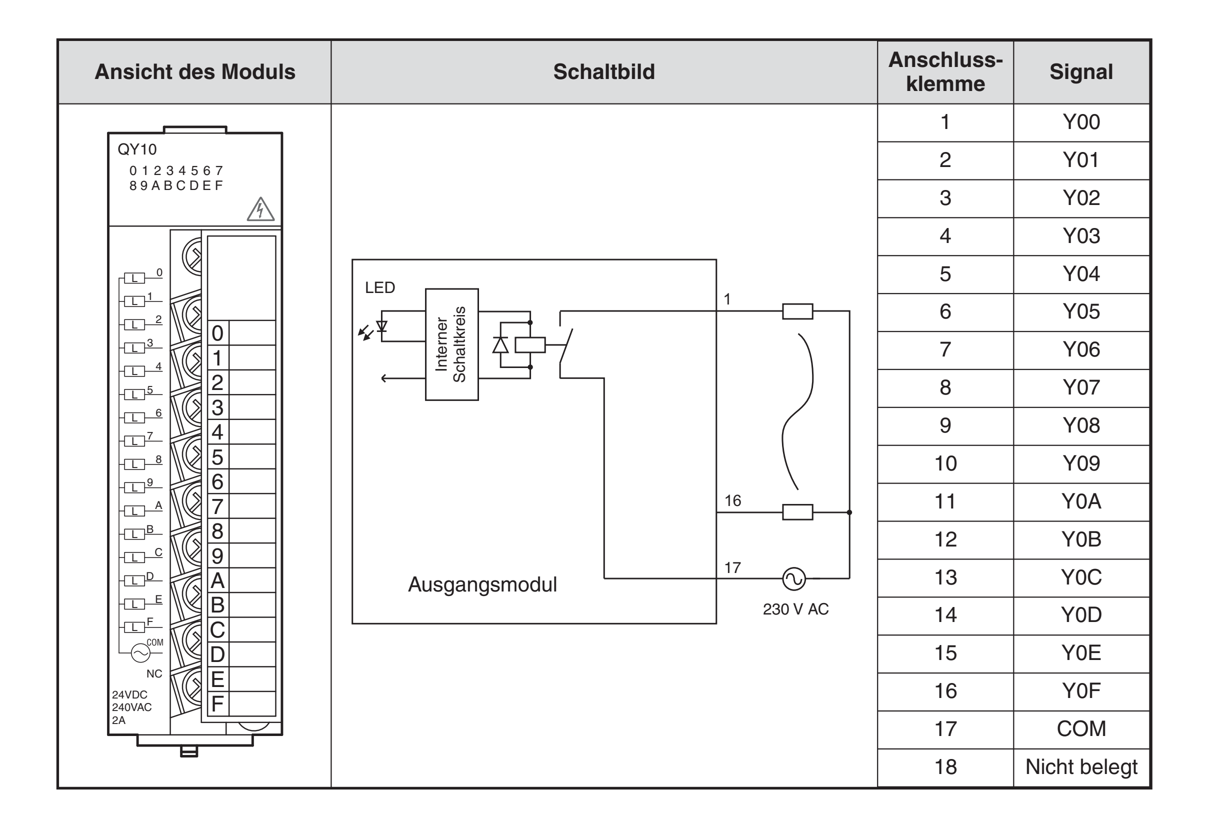
### Relais-Ausgangsmodule

Relais-Ausgangsmodule enthalten pro Ausgang ein Relais, dessen Schaltkontakt wiederum die angeschlossene Lastspannung schaltet. Dadurch wird Trennung zwischen interner Spannung der SPS und externen Lasten erreicht.

Wie auch bei den anderen Ausgangsmodulen wird der Ausgang durch das SPS-Programm gesteuert. Am Ende des Programms werden die SPS-Ausgänge aktualisiert. Das heißt, dass zu diesem Zeitpunkt alle logischen Ausgangszustände, die sich durch das Programm ergeben haben, an die physischen Ausgänge übertragen werden. Ein eingeschalteter Ausgang wird durch eine leuchtende LED angezeigt. Dadurch ist auch eine Kontrolle direkt an der SPS möglich. Ein Relais-Ausgangsmodul hat eine Reaktionszeit von ca. 10 ms.

### Beispiel für ein Relais-Ausgangsmodul

| Merkmal | | Technische Daten |
|---|---|---|
| Bezeichnung des Moduls | | QY10 |
| Ausgänge | | 16 |
| Isolation | | Durch Relais |
| Ausgangsnennspannung/-strom | | 24 V DC 2 A (Ohmsche Last) pro Ausgang<br>240 V AC 2 A (cosφ = 1) pro Ausgang; Max. 8 A je Gruppe |
| Minimale Schaltlast | | 5 V DC, 1 mA |
| Max. Schaltspannung | | 125 V DC / 264 V AC |
| Ansprechzeit | AUS → EIN | ≤10 ms |
| | EIN → AUS | ≤12 ms |
| Lebensdauer der Kontakte | Mechanisch | ≥20 Mio. Schaltungen |
| | Elektrisch | ≥100.000 Schaltungen bei Ausgangsnennspannung/-strom |
| | | ≥100.000 Schaltungen bei 200 V AC, 1,5 A; 240 V AC 1 A (cos φ = 0,7)<br>≥300.000 Schaltungen bei 200 V AC, 0,4 A; 240 V AC 0,3 A (cos φ = 0,7) |
| | | ≥100.000 Schaltungen bei 200 V AC, 1 A; 240 V AC 0,5 A (cos φ = 0,35)<br>≥300.000 Schaltungen bei 200 V AC, 0,3 A; 240 V AC 0,15 A (cos φ = 0,35) |
| | | ≥100.000 Schaltungen bei 24 V DC 1 A; 100 V DC 0,1 A (L/R = 0,7 ms)<br>≥300.000 Schaltungen bei 24 V DC 0,3 A; 100 V DC 0,03 A (L/R = 0,7ms) |
| Max. Schaltfrequenz | | 3600 Schaltungen/Stunde |
| Netzfilter | | — |
| Sicherung | | — |
| Durchschlagfestigkeit | | 2830 V AC Effektivwert für 3 Zyklen (Einsatzhöhe 2000 m) |
| Isolationswiderstand | | ≥10 MΩ (Messung mit Isolationsprüfgerät) |
| Störfestigkeit | | Geprüft mit Störsimulator (Spitzenwert der Störspannung: 1500 V, Einschaltdauer der Störspannung: 1µs, Frequenz der Störspannung: 25 bis 60 Hz) |
| | | Hochfrequente, nicht periodische Störspannung (IEC61000-4-4): 1kV |
| Ausgangsgruppen | | 1 Gruppe mit 16 Ausgängen, Bezugspotential: Anschlussklemme 17 |
| Statusanzeige der Ausgänge | | Eine LED pro Ausgang |
| Anschluss der Verdrahtung | | Klemmblock mit 18 Schraubklemmen (M3 x 6) |
| Empfohlener Leitungsquerschnitt | | 0,3 bis 0,75 mm$^2$, Max. Durchmesser der Drähte: 2,8 mm |
| Interne Stromaufnahme (5 V DC) | | 430 mA |
| Gewicht | | 0,22 kg |



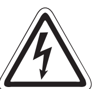

| Ansicht des Moduls | Schaltbild | Anschluss-klemme | Signal |
|---|---|---|---|
| | | 1 | Y00 |
| | | 2 | Y01 |
| | | 3 | Y02 |
| | | 4 | Y03 |
| | | 5 | Y04 |
| | | 6 | Y05 |
| | | 7 | Y06 |
| | | 8 | Y07 |
| | | 9 | Y08 |
| | | 10 | Y09 |
| | | 11 | Y0A |
| | | 12 | Y0B |
| | | 13 | Y0C |
| | | 14 | Y0D |
| | | 15 | Y0E |
| | | 16 | Y0F |
| | | 17 | COM |
| | | 18 | Nicht belegt |

### Triac-Ausgangsmodule

Digitale Triac-Ausgangsmodule schalten Wechselspannungen von 100 bis 240 V. Die Schaltspannung ist durch Optokoppler von der SPS-Versorgungsspannung getrennt. Die Reaktionszeit von Triac-Ausgangsmodulen ist kürzer als die von Relais-Ausgangsmodulen. Zum Einschalten wird 1 ms und zum Ausschalten werden 10 ms benötigt.

Ein Triac kann einen maximalen Strom von 0,6 A schalten. Eine Anlage mit Triac-Ausgangsmodulen muss so ausgelegt werden, dass dieser maximaler Schaltstrom nicht überschritten wird.

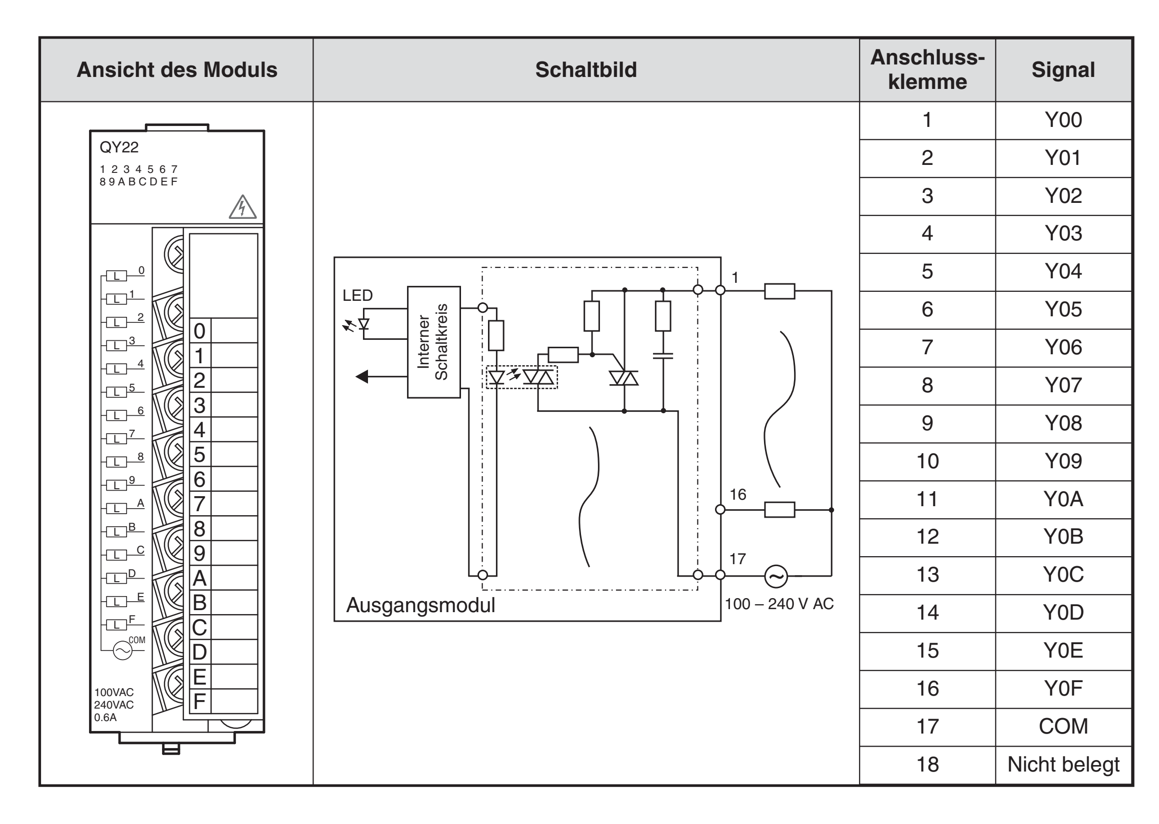
Auch bei ausgeschaltetem Ausgang fließt durch den Triac ein Leckstrom von max. 3 mA. Durch diesen geringen Strom können Meldeleuchten auch bei ausgeschaltetem Ausgang weiter leuchten oder kleine Relais angezogen bleiben.



**GEFAHR:**
***Durch den Leckstrom besteht die Gefahr von elektrischen Schlägen auch bei ausgeschaltetem Triac-Ausgang. Schalten Sie vor Arbeiten an einer elektrischen Anlage immer die Spannung komplett aus.***

### Beispiel für ein Triac-Ausgangsmodul

| Merkmal | | Technische Daten |
|---|---|---|
| Bezeichnung des Moduls | | QY22 |
| Ausgänge | | 16 |
| Isolation | | durch Optokoppler |
| Ausgangsnennspannung/-strom | | 100 – 240 V AC (+20/-15 %), 0,6 A pro Ausgang, 4,8 A pro Modul |
| Minimale Schaltlast | | 24 V AC, 100 mA; 100 V AC, 25 mA, 240 V AC, 25 mA |
| Max. Einschaltstromspitze | | 20 A |
| Leckstrom bei ausgeschaltetem Ausgang | | ≤ 3 mA bei 120 V AC, 60 Hz<br>≤ 1,5 mA bei 240 V AC, 60 Hz |
| Max. Spannungsabfall bei eingeschaltetem Ausgang | | 1,5 V |
| Ansprechzeit | AUS → EIN | 0,5 x Periodendauer + max. 1 ms |
| | EIN → AUS | 0,5 x Periodendauer + max. 1 ms |
| Netzfilter | | RC-Element |
| Sicherung | | — |
| Durchschlagfestigkeit | | 2830 V AC Effektivwert für 3 Zyklen (Einsatzhöhe 2000 m) |
| Isolationswiderstand | | ≥10 MΩ (Messung mit Isolationsprüfgerät) |
| Störfestigkeit | | Geprüft mit Störsimulator (Spitzenwert der Störspannung: 1500 V, Einschaltdauer der Störspannung: 1μs, Frequenz der Störspannung: 25 bis 60 Hz) |
| | | Hochfrequente, nicht periodische Störspannung (IEC61000-4-4): 1kV |
| Ausgangsgruppen | | 1 Gruppe mit 16 Ausgängen, Bezugspotential: Anschlussklemme 17 |
| Statusanzeige der Ausgänge | | Eine LED pro Ausgang |
| Anschluss der Verdrahtung | | Klemmblock mit 18 Schraubklemmen (M3 x 6) |
| Empfohlener Leitungsquerschnitt | | 0,3 bis 0,75 mm², Max. Durchmesser der Drähte: 2,8 mm |
| Interne Stromaufnahme (5 V DC) | | 250 mA (Alle Ausgänge sind eingeschaltet.) |
| Gewicht | | 0,40 kg |




| Ansicht des Moduls | Schaltbild | Anschlussklemme | Signal |
|---|---|---|---|
| | | 1 | Y00 |
| | | 2 | Y01 |
| | | 3 | Y02 |
| | | 4 | Y03 |
| | | 5 | Y04 |
| | | 6 | Y05 |
| | | 7 | Y06 |
| | | 8 | Y07 |
| | | 9 | Y08 |
| | | 10 | Y09 |
| | | 11 | Y0A |
| | | 12 | Y0B |
| | | 13 | Y0C |
| | | 14 | Y0D |
| | | 15 | Y0E |
| | | 16 | Y0F |
| | | 17 | COM |
| | | 18 | Nicht belegt |

### Transistor-Ausgangsmodule

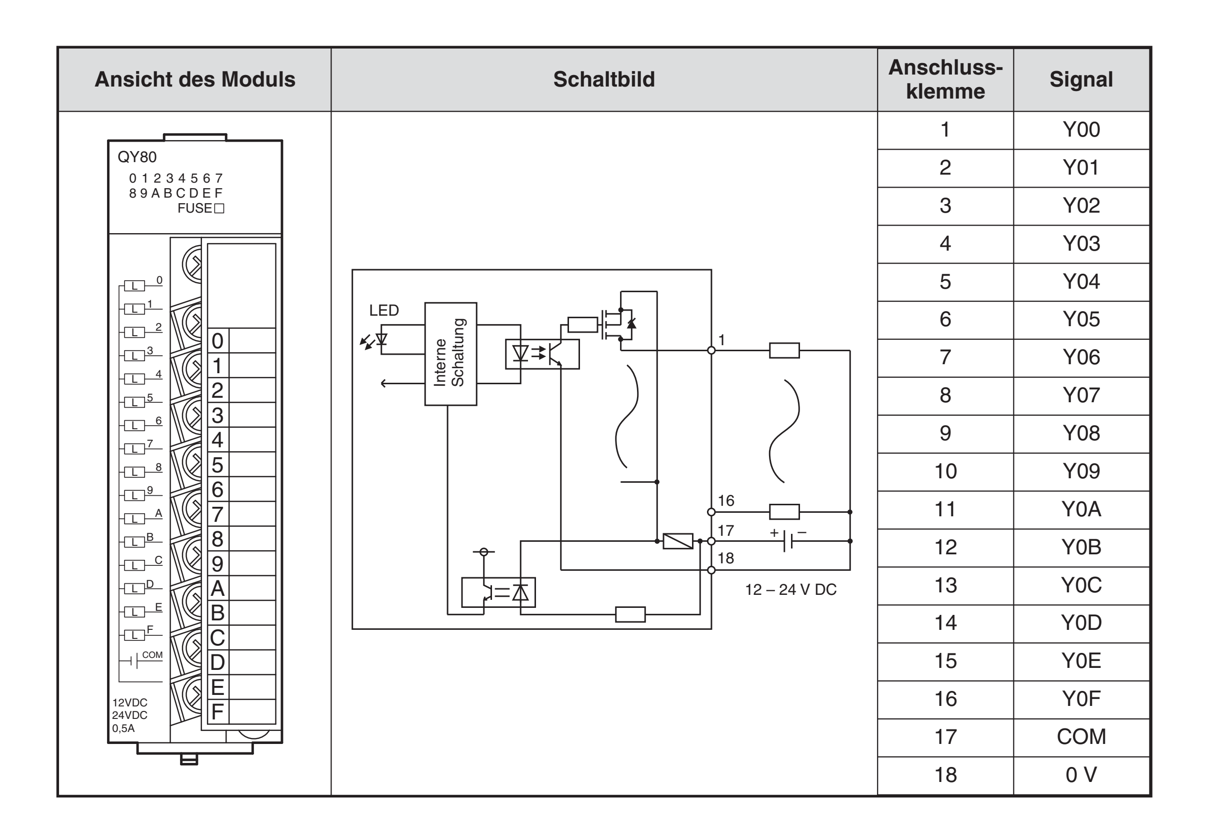
Auch bei Transistor-Ausgangsmodulen sind die Schaltspannung und die Versorgungsspannung der SPS durch Optokoppler isoliert.

Ein Transistor-Ausgangsmodul benötigt nur 1 ms, um einen Ausgang zu schalten. Die technischen Daten wie zum Beispiel die Schaltströme können den Handbüchern der Module oder der Installationsanleitung zu den Ein- und Ausgabemodulen (Art.-Nr. 141758) entnommen werden.

Im MELSEC System Q sind plus- oder minusschaltende Ausgangsmodule erhältlich.

### Beispiel für ein plusschaltendes Ausgangsmodul

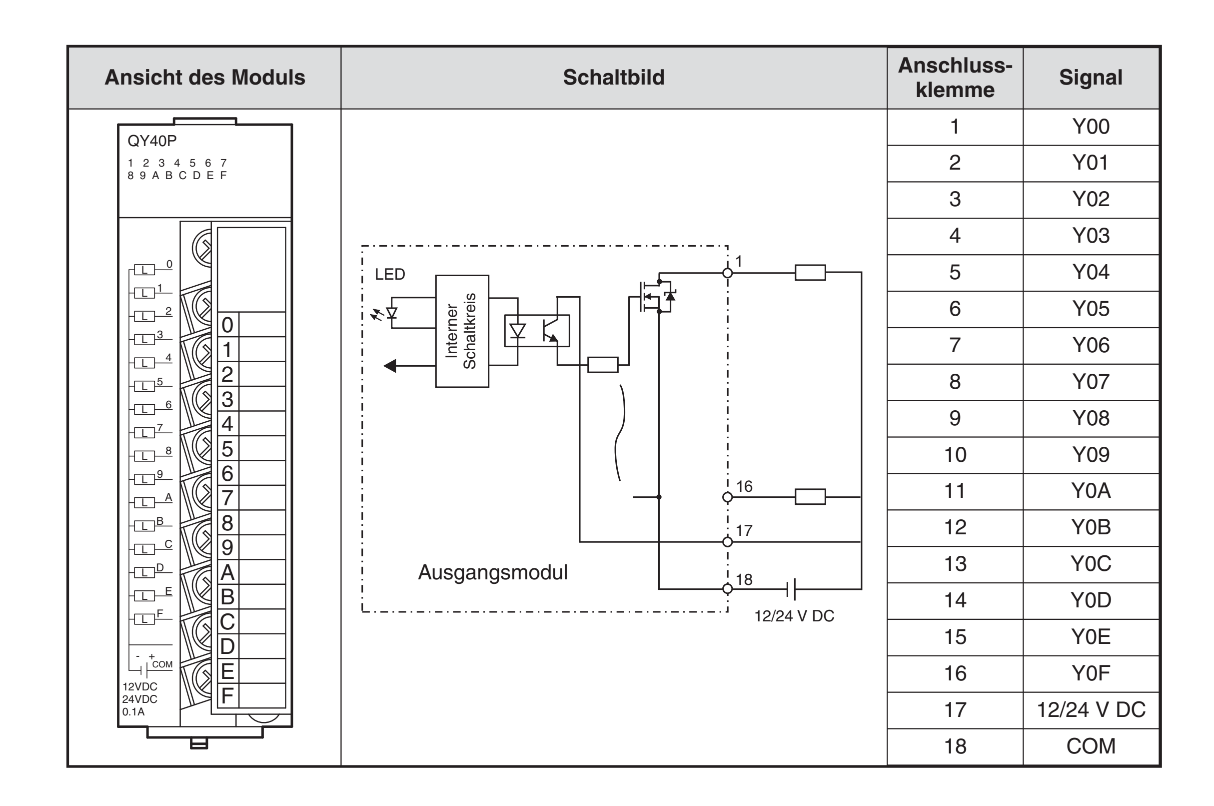
| Merkmal | | Technische Daten |
|---|---|---|
| Bezeichnung des Moduls | | QY80 |
| Ausgänge | | 16 |
| Isolation | | durch Optokoppler |
| Ausgangsnennspannung | | 12 bis 24 V DC (+20/-15%) |
| Ausgangsspannungsbereich | | 10,2 bis 28,8 V DC |
| Max. Schaltbedingung | | 0,5 A pro Ausgang, 4 A pro Gruppe |
| Max. Einschaltstromspitze | | 4 A für 10 ms |
| Leckstrom bei ausgeschaltetem Ausgang | | ≤0,1 mA |
| Spannungsabfall bei eingeschaltetem Ausgang | | Typisch 0,2 V DC bei 0,5 A, max. 0,3 V bei 0,5 A |
| Ansprechzeit | AUS → EIN | ≤1 ms |
| | EIN → AUS | ≤1 ms (bei Nennschaltbedingungen und ohmscher Belastung) |
| Netzfilter | | Z-Diode |
| Sicherung | | 6,7 A; nicht austauschbar |
| Anzeige einer defekten Sicherung | | Durch Einschalten einer LED und Signal an die CPU |
| Versorgung des Moduls | Spannung | 12 bis 24 V DC (+20/-15%, Welligkeit 5%) |
| | Strom | 20 mA (bei 24 V DC und wenn alle Ausgänge geschaltet sind) |
| Durchschlagfestigkeit | | 560 V AC Effektivwert für 3 Zyklen (Einsatzhöhe 2000 m) |
| Isolationswiderstand | | ≥10 MΩ (Messung mit Isolationsprüfgerät) |
| Störfestigkeit | | Geprüft mit Störsimulator (Spitzenwert der Störspannung: 500 V, Einschaltdauer der Störspannung: 1 µs, Frequenz der Störspannung: 25 bis 60 Hz) |
| | | Hochfrequente, nicht periodische Störspannung (IEC61000-4-4): 1kV |
| Ausgangsgruppen | | 1 Gruppe mit 16 Ausgängen, Bezugspotential: Anschlussklemme 17 |
| Statusanzeige der Ausgänge | | Eine LED pro Ausgang |
| Anschluss der Verdrahtung | | Klemmblock mit 18 Schraubklemmen (M3 x 6) |
| Empfohlener Leitungsquerschnitt | | 0,3 bis 0,75 $mm^2$, Max. Durchmesser der Drähte: 2,8 mm |
| Interne Stromaufnahme (5 V DC) | | 80 mA |
| Gewicht | | 0,17 kg |




| Ansicht des Moduls | Schaltbild | Anschluss-klemme | Signal |
|---|---|---|---|
| | | 1 | Y00 |
| | | 2 | Y01 |
| | | 3 | Y02 |
| | | 4 | Y03 |
| | | 5 | Y04 |
| | | 6 | Y05 |
| | | 7 | Y06 |
| | | 8 | Y07 |
| | | 9 | Y08 |
| | | 10 | Y09 |
| | | 11 | Y0A |
| | | 12 | Y0B |
| | | 13 | Y0C |
| | | 14 | Y0D |
| | | 15 | Y0E |
| | | 16 | Y0F |
| | | 17 | COM |
| | | 18 | 0 V |

### Beispiel für ein minusschaltendes Ausgangsmodul

| Merkmal | | Technische Daten |
|---|---|---|
| Bezeichnung des Moduls | | QY40P |
| Ausgänge | | 16 |
| Isolation | | durch Optokoppler |
| Ausgangsnennspannung | | 12 bis 24 V DC (+20/-15%) |
| Ausgangsspannungsbereich | | 10,2 bis 28,8 V DC |
| Max. Schaltbedingung | | 0,1 A pro Ausgang, 1,6 A pro Gruppe |
| Max. Einschaltstromspitze | | 0,7 A für 10 ms |
| Leckstrom bei ausgeschaltetem Ausgang | | ≤0,1 mA |
| Spannungsabfall bei eingeschaltetem Ausgang | | Typisch 0,1 V DC bei 0,1 A, max. 0,2 V bei 0,1 A |
| Ansprechzeit | AUS → EIN | ≤1 ms |
| | EIN → AUS | ≤1 ms (bei Nennschaltbedingungen und ohmscher Belastung) |
| Netzfilter | | Z-Diode |
| Sicherung | | — |
| Versorgung des Moduls | Spannung | 12 bis 24 V DC (+20/-15%, Welligkeit 5%) |
| | Strom | 10 mA (bei 24 V DC und wenn alle Ausgänge geschaltet sind) |
| Durchschlagfestigkeit | | 560 V AC Effektivwert für 3 Zyklen (Einsatzhöhe 2000 m) |
| Isolationswiderstand | | ≥10 MΩ (Messung mit Isolationsprüfgerät) |
| Störfestigkeit | | Geprüft mit Störsimulator (Spitzenwert der Störspannung: 500 V, Einschaltdauer der Störspannung: 1µs, Frequenz der Störspannung: 25 bis 60 Hz) |
| | | Hochfrequente, nicht periodische Störspannung (IEC61000-4-4): 1kV |
| Ausgangsgruppen | | 1 Gruppe mit 16 Ausgängen, Bezugspotential: Anschlussklemme 18 |
| Statusanzeige der Ausgänge | | Eine LED pro Ausgang |
| Anschluss der Verdrahtung | | Klemmblock mit 18 Schraubklemmen (M3 x 6) |
| Empfohlener Leitungsquerschnitt | | 0,3 bis 0,75 mm$^2$, Max. Durchmesser der Drähte: 2,8 mm |
| Interne Stromaufnahme (5 V DC) | | 65 mA |
| Gewicht | | 0,16 kg |




| Anschluss-klemme | Signal |
|---|---|
| 1 | Y00 |
| 2 | Y01 |
| 3 | Y02 |
| 4 | Y03 |
| 5 | Y04 |
| 6 | Y05 |
| 7 | Y06 |
| 8 | Y07 |
| 9 | Y08 |
| 10 | Y09 |
| 11 | Y0A |
| 12 | Y0B |
| 13 | Y0C |
| 14 | Y0D |
| 15 | Y0E |
| 16 | Y0F |
| 17 | 12/24 V DC |
| 18 | COM |

# 3.6 Sondermodule

## 3.6.1 Analogmodule

Bei der Automatisierung eines Prozesses müssen häufig analoge Größen wie beispielsweise Temperaturen, Drücke oder Füllstände gemessen und gesteuert oder geregelt werden. Dazu werden besondere Analogmodule benötigt.

Grundsätzlich kann unterschieden werden zwischen

- Analogeingangsmodulen und
- Analogausgangsmodulen.

Analogeingangsmodule können Ströme, Spannungen oder Temperaturen erfassen. Analogausgangsmodule dienen zur Ausgabe von Strömen oder Spannungen.

**Auswahlkriterien für Analogmodule**

Das MELSEC System Q bietet eine große Auswahl an Analogmodulen. Zur Lösung einer bestimmten Automatisierungsaufgabe muss daher eine Auswahl getroffen werden. Die Hauptkriterien bei dieser Entscheidung sind:

- Auflösung

  Die „Auflösung“ gibt an, welcher kleinste physikalische Wert von einem Analogmodul erfasst bzw. ausgegeben werden kann.

  Bei Analogeingangsmodulen versteht man unter Auflösung die Änderung der Spannung, des Stroms oder der Temperatur am Eingang, die eine Erhöhung oder Verminderung des digitalen Ausgangswerts um „1“ ergibt.

  Bei Analogausgangsmodulen bezeichnet die Auflösung die Änderung des Spannungs- oder Stromwerts am Ausgang des Moduls bei einer Erhöhung oder Verminderung des digitalen Eingangswerts um „1“.

  Die Auflösung ist durch den internen Aufbau der Analogmodule vorgegeben und hängt davon ab, wie viele Bits zur Speicherung des digitalen Werts benötigt werden. Wird zum Beispiel eine Spannung von 10 V mit einem 12-Bit-A/D-Wandler erfasst, wird die Spannung in 4096 Schritte unterteilt ($2^{12}$ = 4096, siehe Abschnitt 4.3). Dadurch ergibt sich eine Auflösung von 10 V/4096 = 2,5 mV.

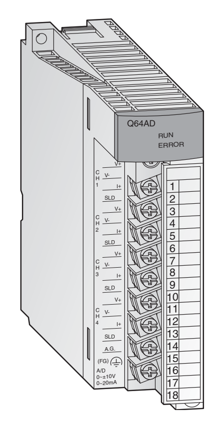
- Anzahl der analogen Ein- oder Ausgänge

  Die Ein- oder Ausgänge eines Analogmoduls werden auch als Kanäle bezeichnet. Entsprechend der Anzahl der benötigten Kanäle können zum Beispiel Analogeingangsmodule mit 4 oder 8 Kanälen gewählt werden.

**Analogeingangsmodule**

Analogeingangsmodule wandeln einen gemessenen analogen Wert (z. B. 10 V) in einen digitalen Wert (z. B. 4000), der von der SPS weiter verarbeitet werden kann. Dieser Vorgang wird als Analog/Digitalwandlung oder auch kurz als A/D-Wandlung bezeichnet.

Während Temperaturen mit Analogmodulen des MELSEC System Q direkt erfasst werden können, müssen andere physikalische Signale, wie z. B. Drücke oder Durchflussmengen, erst in Strom- oder Spannungswerte gewandelt werden, bevor sie von der SPS verarbeitet werden können. Diese Wandlung wird von Messaufnehmern übernommen, die an ihren Ausgängen genormte Signale zur Verfügung stellen (zum Beispiel 0 bis 10 V oder 4 bis 20 mA.) Die Messung eines Stromes hat den Vorteil, dass der Messwert nicht durch die Länge der Leitungen oder durch Übergangswiderstände beeinflusst wird.



Die Analogeingangsmodule des System Q vereinen eine hohe Auflösung von bis zu 0,333 mV bzw.1,33 mA mit einer extrem kurzen Wandlungszeit von nur 80 µs pro Eingang.

Der Anschluss der Eingangssignale erfolgt bei allen Modulen über eine abnehmbare Klemmleiste mit Schraubklemmen.

| Eingangsart | Nenneingangsbereich | Einstellbarer Eingangsbereich | Anzahl der Eingänge | Modul |
|---|---|---|---|---|
| Spannung | -10 bis +10 V | 1 bis 5 V<br>0 bis 5 V<br>0 bis 10 V<br>-10 bis +10 V | 8 | Q68ADV |
| Strom | 0 bis 20 mA | 0 bis 20 mA<br>4 bis 20 mA | 8 | Q68ADI |
| Spannung oder Strom (für jeden Eingang separat wählbar) | -10 bis +10 V<br>0 bis 20 mA | Wie bei Q68ADV und Q68ADI | 4 | Q64AD |

### Analogeingangsmodule zur Messung von Temperaturen

Zur Erfassung der Temperatur werden entweder Widerstandsthermometer oder Thermoelemente verwendet.

- Widerstandsthermometer

  Bei dieser Art der Temperaturmessung wird der Widerstand eines Temperaturfühlers gemessen, der sich bei steigender Temperatur vergrößert. Weit verbreitet sind Temperatursensoren aus Platin. Bei 0 °C hat das Platinelement einen Widerstand von 100 Ω. (Daher auch die Bezeichnung Pt100.) Die Widerstandssensoren werden nach dem Dreileiterverfahren angeschlossen. Dadurch beeinflusst der Widerstand der Anschlussleitungen nicht das Messergebnis.

  Der Messbereich von Pt100-Widerstandsthermometern reicht von -200 °C bis 600 °C, hängt aber auch vom verwendeten Temperaturerfassungsmodul ab.

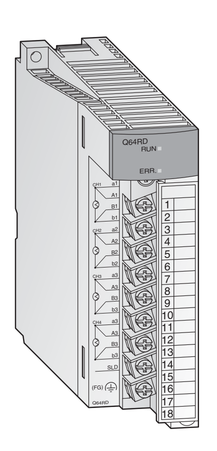
  Widerstandsthermometer können auch aus einer Nickellegierung hergestellt werden (Ni100). Der Temperaturmessbereich ist in diesem Fall kleiner (-60 °C bis 180 °C).

- Thermoelemente

  Bei dieser Temperaturmessmethode wird ausgenutzt, dass bei der Verbindung unterschiedlicher Metalle durch Temperatur eine Spannung erzeugt wird. Dieses Prinzip der Temperaturmessung beruht also auf eine Spannungsmessung.

  Es gibt verschiedene Arten von Thermoelementen. Sie unterscheiden sich in der Thermospannung und den erfassbaren Temperaturbereichen. Die Werkstoffkombination ist genormt und wird durch eine Typenbezeichnung angegeben. Häufig verwendete Thermoelemente sind die Typen J und K. Thermoelemente vom Typ K bestehen aus einer NiCr-Ni-Werkstoffkombination. Zur Herstellung von Thermoelementen vom Typ J wird Eisen (Fe) mit eine Kupfer/Nickellegierung (CuNi) kombiniert. Die Thermoelemente unterscheiden sich außer im Aufbau durch den erfassbaren Temperaturbereich.

  Mit Thermoelementen können Temperaturen von -200 °C bis 1200 °C gemessen werden.




**Besondere Merkmale**

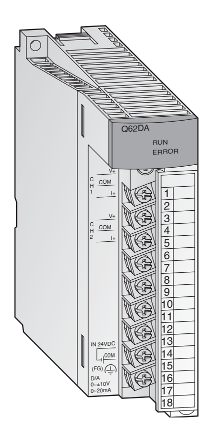
- 4 Kanäle zur Temperaturerfassung pro Modul
- Ein Drahtbruch des Temperatursensors wird der SPS-CPU durch das Modul angezeigt.
- Mittelwertbildung über die Zeit oder über Messzyklen kann parametriert werden
- Fehlerkorrektur durch Einstellung von Offset und Verstärkung
- Alarmausgabe bei Grenzwertüberschreitung
- Standardmäßig Potentialtrennung zwischen Prozess und Steuerung durch Optokoppler. Zusätzliche Potentialtrennung der Kanäle untereinander bei den Modulen Q64TDV-GH und Q64RD-G.

| Temperatursensor | Temperaturmessbereich | Max. Auflösung | Modul |
|---|---|---|---|
| Widerstandsthermometer (Pt100, JPt100) | Pt100: -200 bis 850 °C, JPt 100: -180 bis 600 °C | 0,025 °C | Q64RD |
| Widerstandsthermometer (Pt100, JPt100, Ni100) | Pt100: -200 bis 850 °C, JPt 100: -180 bis 600 °C, Ni100: -60 bis 180 °C | 0,025 °C | Q64RD-G |
| Thermoelemente vom Typ K, E, J, T, B, R, S oder N | Abhängig vom verwendeten Thermoelement | B, R, S, N: 0,3 °C; K, E, J, T: 0,1 °C | Q64TD |
| | | B: 0,7 °C; R, S: 0,8 °C; K, T: 0,3 °C; E,T: 0,2 °C; J: 0,1 °C; N: 0,4 °C; Spannungsmessung: 4 µV | Q64TDV-GH |

**Analog-Ausgangsmodule**

Analogausgangsmodule wandeln einen digitalen Wert, der aus dem SPS-Grundgerät stammt, in ein analoges Strom- oder Spannungssignal, mit dem dann externe Geräte gesteuert werden können (Digital-/Analogwandlung oder kurz D/A-Wandlung).

Die analogen Ausgangssignale der Analogmodule des MELSEC System Q entsprechen dem Industrie-Standard von -10V bis +10 V oder 4 bis 20 mA.




Bei einer extrem kurzen Wandlungszeit von nur 80 µs pro Ausgang wird eine Auflösung von bis zu 0,333 mV bzw. 0,83 µA erreicht. Die kurzschlussfesten Ausgänge sind durch Optokoppler von der Steuerung isoliert.

Bei allen Modulen erfolgt der Anschluss über eine abnehmbare Klemmleiste mit Schraubklemmen.

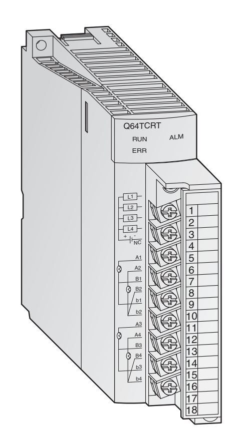
| Ausgangsart | Nennausgangsbereich | Einstellbarer Ausgangsbereich | Anzahl der Ausgänge | | |
|---|---|---|---|---|---|
| | | | 2 | 4 | 8 |
| Spannung oder Strom (für jeden Ausgang separat wählbar) | -10 bis +10 V<br>0 bis 20 mA | 1 bis 5 V<br>-10 bis +10 V<br>0 bis 20 mA<br>4 bis 20 mA | Q62DA | Q64DA | |
| Spannung | -10 bis +10 V | -10 bis +10 V | | | Q68DAV |
| Strom | 0 bis 20 mA | 0 bis 20 mA<br>4 bis 20 mA | | | Q68DAI |

## 3.6.2 Temperaturregelmodule mit PID-Algorithmus

Die Temperaturregelmodule ermöglichen die Temperaturregelung, ohne dass für die Regelungsaufgaben die CPU der SPS belastet wird.



**Besondere Merkmale:**



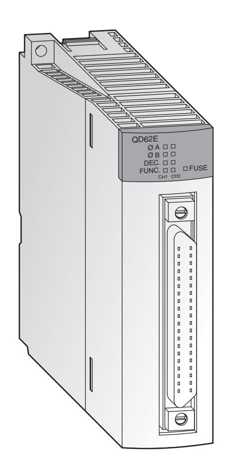

- 4 Kanäle zur Temperaturerfassung und 4 PID-Regelkreise pro Modul
- Temperaturmessung entweder mit Pt100-Widerstandsthermometern (Q64TCRT und Q64TCRTBW) oder mit Thermoelementen (Q64TCTT und Q64TCTTBW)
- Integrierte Drahtbrucherkennung für die Heizung bei den Modulen Q64TCRTBW und Q64TCTTBW
- Optimierung der Regelung durch Autotuning
- Transistorausgang für die Ansteuerung des Stellgliedes

## 3.6.3 High-Speed-Zählermodule

Die Zählermodule QD62E, QD62 und QD62D erfassen Impulse, deren Frequenz für normale Eingangsmodule zu hoch ist.



**Besondere Merkmale:**

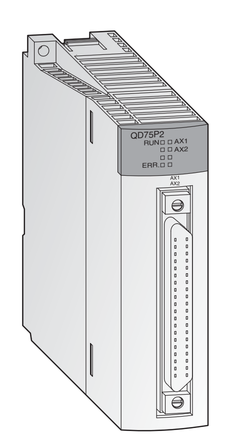



- Max. Zählfrequenz bis zu 500 kHz
- Eingang für Inkrementaldrehgeber mit automatischer Vor- und Rückwärtserkennung
- Zählwertvorgabe und Funktionsauswahl über digitale Eingänge
- 32-Bit-Zählbereich mit Vorzeichen (-2 147 483 648 bis +2 147 483 647)
- Als auf- oder abwärtszählender Zähler oder Ringzähler einsetzbar
- Alle Module bieten zwei Zählereingänge.
- Pro Zählkanal stehen 2 digitale Ausgänge zur Verfügung, die abhängig vom Zählwert geschaltet werden

Alle Module werden über eine 40-polige Steckverbindung angeschlossen.

### 3.6.4 Positioniermodule

In Verbindung mit Schrittmotoren oder Servoverstärkern können die Positioniermodule QD75P1, QD75P2 und QD75P4 zur Positionierung oder Geschwindigkeitssteuerung eingesetzt werden.

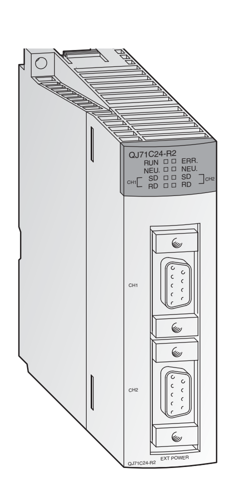



**Besondere Merkmale:**

- Steuerung von bis zu vier Achsen linearinterpolierend (QD75P4) oder zwei Achsen zirkularinterpolierend (QD75P2 und QD75P4)
- Speicherung von bis zu 600 Positionsdaten im Flash-ROM
- Als Einheiten bei der Positionierung können Impulse, µm, Inch oder Winkelgrade vorgegeben werden.
- Parametrierung und Positionsdatenvorgabe über das SPS-Programm oder mit Hilfe der Programmier-Software GX Configurator QP.

### 3.6.5 Schnittstellenmodule zur seriellen Übertragung

Die Module QJ71C24 und QJ71C24-R2 dienen zur Kommunikation mit Peripheriegeräten. Dabei werden standardisierte serielle Schnittstellen verwendet.

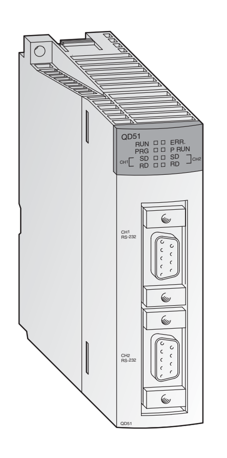



**Besondere Merkmale:**

- Zwei RS232C-Schnittstellen (bei QJ71C24-R2 ) oder eine RS422/485- und eine RS232C-Schnittstelle (bei QJ71C24)
- Übertragungsgeschwindigkeiten von bis zu 115200 Baud
- Zugriffsmöglichkeit auf die Daten der SPS durch übergeordnete PCs mit Visualisierungs- oder Monitorsoftware
- Der Anschluss eines Druckers ist möglich.
- Integrierter Speicher zur Ablage von Qualitäts-, Produktions- oder Alarmdaten, die nach Bedarf übertragen werden
- Ein freies Protokoll zum Datenaustausch kann definiert werden
- Die Programmierung der SPS über die Schnittstellenmodule ist möglich.

## 3.6.6 BASIC-programmierbare Schnittstellenmodule

Die Module QD51S-R24 und QD51 arbeiten unabhängig von der CPU der SPS ein eigenes Programm ab, das in AD51H-Basic geschrieben wird. So können Daten mit Peripheriegeräten ausgetauscht werden, ohne dass die SPS-CPU belastet wird.




**Besondere Merkmale:**

- Entweder zwei RS232C-Schnittstellen (bei QD51) oder eine RS422/485- und eine RS232C-Schnittstelle (bei QD51S-R24)
- Übertragungsgeschwindigkeiten von bis zu 38400 Baud
- Auf Operanden innerhalb der SPS-CPU und Pufferspeichern von Sondermodulen kann zugegriffen werden.
- Über die Schnittstellenmodule kann die Betriebsart der SPS-CPU ferngesteuert geändert werden (RUN/STOP-Umschaltung)

## 3.7 Netzwerke und Netzwerkmodule

### 3.7.1 Vernetzung auf allen Ebenen

In komplexen oder weit verzweigten Anwendungen, aber auch zur Realisierung von dezentralen Ein- und Ausgängen oder zur Visualisierung von Prozessen nimmt die Kommunikation zwischen Steuerungen, zentralen Rechnern, Bediengeräten oder anderen externen Geräten einen hohen Stellenwert ein.

Mitsubishi Electric bietet optimale Lösungen auf der Basis von drei Netzwerkebenen:

- Fertigungsebene
- Steuerungsebene
- Leitebene

**Fertigungsebene**

Die unterste Ebene in einer Netzwerkarchitektur bildet ein sogenannter Feldbus, über den ein Datenaustausch zwischen SPS und den verschiedensten Geräten, wie z. B. dezentralen Ein- und Ausgängen, Frequenzumrichtern oder Bediengeräten möglich ist.

Bei einem herkömmlichen zentralen Aufbau wird jeder Sensor direkt mit der SPS im Schaltraum verbunden. Mit einem dezentralen Aufbau dagegen kann der Aufwand für die Verdrahtung reduziert werden. Zusätzlich zu digitalen Daten lassen sich auch Informationen von und zu Anzeigeeinrichtungen, Bar-Code-Lesern, Messvorrichtungen und Personal-Computern übertragen. Dadurch kann der Wartungsaufwand verringert werden, weil der Zustand der Anlage zentral überwacht werden kann. Die hohe Geschwindigkeit und Leistungsfähigkeit werden in Verbindung mit einer SPS des MELSEC System Q weiter gesteigert, gleichzeitig ist eine einfache Anwendung sichergestellt.

**Steuerungsebene**

Die mittlere Ebene in der Netzwerkarchitektur bildet ein Netzwerk zur Verbindung von Geräten, die Prozesse steuern, wie z. B. SPS und CNC.

Weil Daten ausgetauscht werden, die direkten Einfluss auf Prozesse oder die Bewegung von Maschinen haben, ist bei dieser Art von Netzwerken wichtig, dass die Daten unverzögert übermittelt werden. Die Netzwerke MELSECNET/10 und MELSECNET/H zeichnen sich durch Echtzeitfähigkeit, einfache Parametrierung und Ausfallsicherheit durch den Aufbau der Verbindung als redundanter Ring aus.

**Leitebene**

Die Leitebene bildet die oberste Ebene in einer Netzwerkarchitektur für Produktionsstätten. Zwischen der SPS und dem Leitrechner werden Produktionsdaten, Daten zur Qualitätssicherung, Zustände der Produktionsanlage etc. über das weitverbreitete ETHERNET ausgetauscht. Am Ethernet können nicht nur PC's mit den Betriebssystemen Windows oder UNIX, sondern auch viele Geräte zur Anlagenautomatisierung angeschlossen werden. Die MELSEC-SPS verfügen über verschiedene Funktionen, um das Ethernet optimal und einfach zu nutzen.

Zusätzlich können die Netzwerke in

- Offene Netzwerke

und in

- MELSEC-Netzwerke

eingeteilt werden.

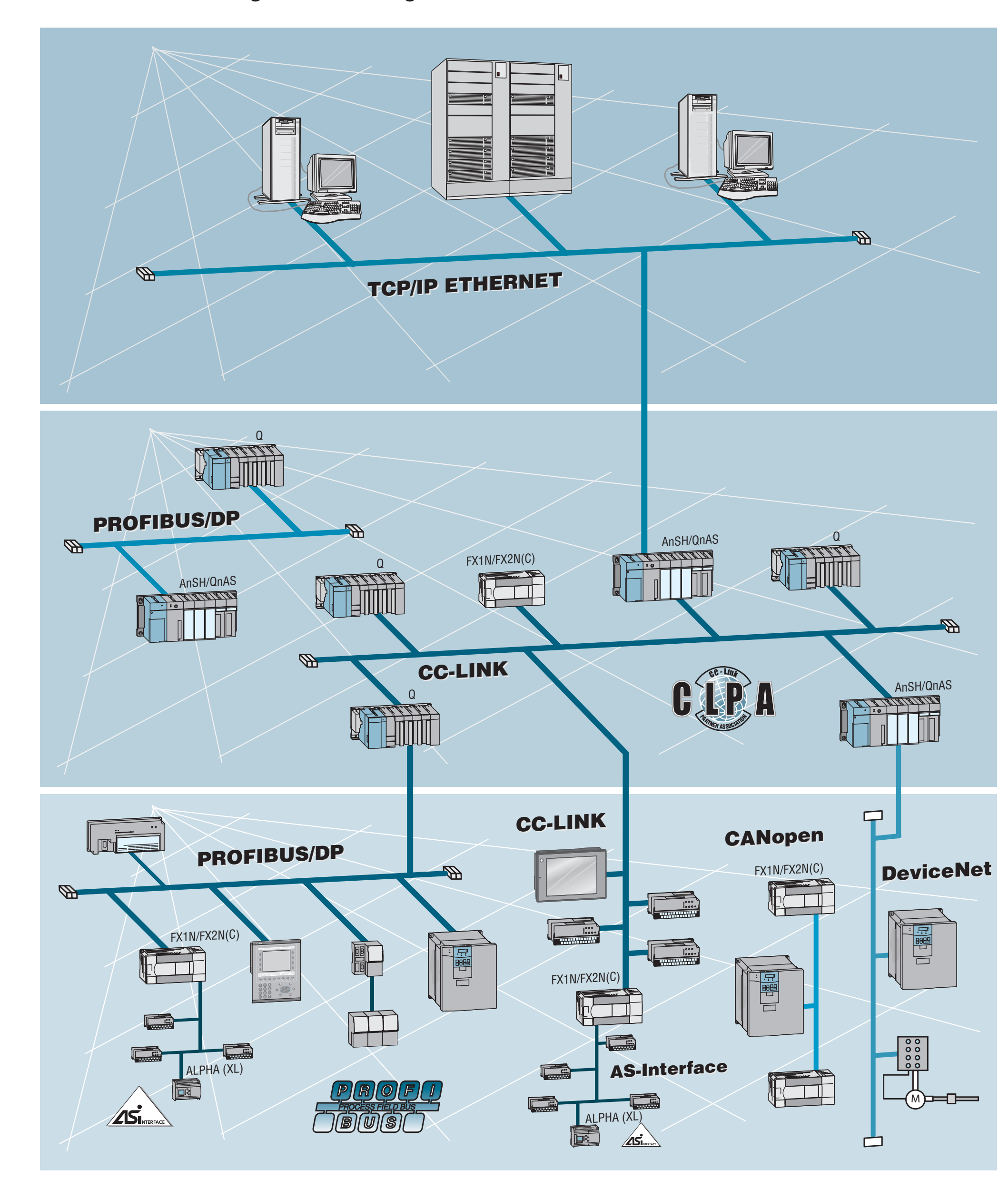
## 3.7.2 Offene Netzwerke

Offene Netzwerke sind herstellerübergreifend, das heißt, dass diese Netzwerke auch von anderen Herstellern verwendet werden und dadurch der Datenaustausch zwischen einer MELSEC SPS und Fremdgeräten ermöglicht wird.

**Leitebene**
ETHERNET

**Steuerungsebene**
PROFIBUS/DP
CC-Link

**Fertigungsebene**
PROFIBUS/DP
DeviceNet
AS-Interface
CC-Link
CANopen




### ETHERNET

Eines der weitverbreitesten Netzwerke ist das ETHERNET. Es verbindet die Bürowelt mit den Steuerungssystemen. ETHERNET ist eine Plattform für die verschiedensten Transportprotokolle, und mit dem exakt auf das ETHERNET zugeschnittene TCP/IP-Protokoll ist ein schneller Datenaustausch zwischen Prozessvisualisierung und den MELSEC SPS möglich.

TCP/IP bildet logische Punkt-zu-Punkt-Verbindungen zwischen zwei ETHERNET-Teilnehmern. Darauf basierend kann z. B. beim MELSEC System Q eine Prozessvisualisierung bis zu 960 Datenworte pro Anfrage anfordern.

### PROFIBUS/DP

Über ein PROFIBUS/DP-Netzwerk ist ein äußerst schneller Datenaustausch mit den verschiedensten Geräten möglich, wie z. B.:

- dezentrale digitale oder analoge Ein-/Ausgänge
- Frequenzumrichter
- Bediengeräte
- verschiedenste andere Geräte von Fremdherstellern

Zur Datenübertragung wird beim PROFIBUS/DP eine kostengünstige 2-Draht-Leitung verwendet.

### CC-Link

Über das Netzwerk MELSEC CC-Link ist ein schneller Datenaustausch mit den verschiedensten Geräten möglich. In das Netzwerk sind z. B. folgende Produkte von MITSUBISHI ELECTRIC integrierbar:

- SPS-Systeme
- dezentrale digitale oder analoge Ein-/Ausgänge
- Positioniermodule
- Frequenzumrichter
- Bediengeräte
- Industrieroboter
- Fremdgeräte wie z. B. Bar-Code-Lesegeräte

Verschiedenste Daten wie z. B. digitale und analoge Daten lassen sich problemlos transportieren. Zusätzlich zu der zyklischen Übertragung von Daten können in einem CC-Link-System auch Mitteilungen versendet werden. Hierdurch wird eine Kommunikation mit dezentralen Geräten wie Anzeigeeinheiten, Bar-Code-Lesern, Messvorrichtungen, Personal-Computern und SPS-Systemen (mit bis zu 24 CPUs) ermöglicht.

### DeviceNet

Das DeviceNet ist eine kostengünstige Lösung für die Vernetzung von Geräten in der Fertigungsebene. Bis zu 64 Geräte einschließlich Master können in einem Netzwerk betrieben werden.

### AS-Interface

Das AS-Interface ist ein internationaler Standard für die unterste Feldbusebene. Das Netzwerk ist universell einsetzbar, sehr flexibel und besonders einfach zu installieren. Angeschlossen werden können **A**ktoren, wie Ventile oder Anzeigegeräte, und **S**ensoren, daher auch die Bezeichnung AS-i.

### CANopen

CANopen ist eine „offene“ Lösung des von den Mitgliedern der internationalen Anwender- und Herstellervereinigung entwickelten *Controller Area Network* (CAN).

CANopen-Netzwerke werden zur Verbindung von Sensoren, Aktoren und Steuergeräten im Maschinenbau, der Medizintechnik, in der Schifffahrt, in Eisen- und Straßenbahnen sowie in Nutzfahrzeugen eingesetzt.

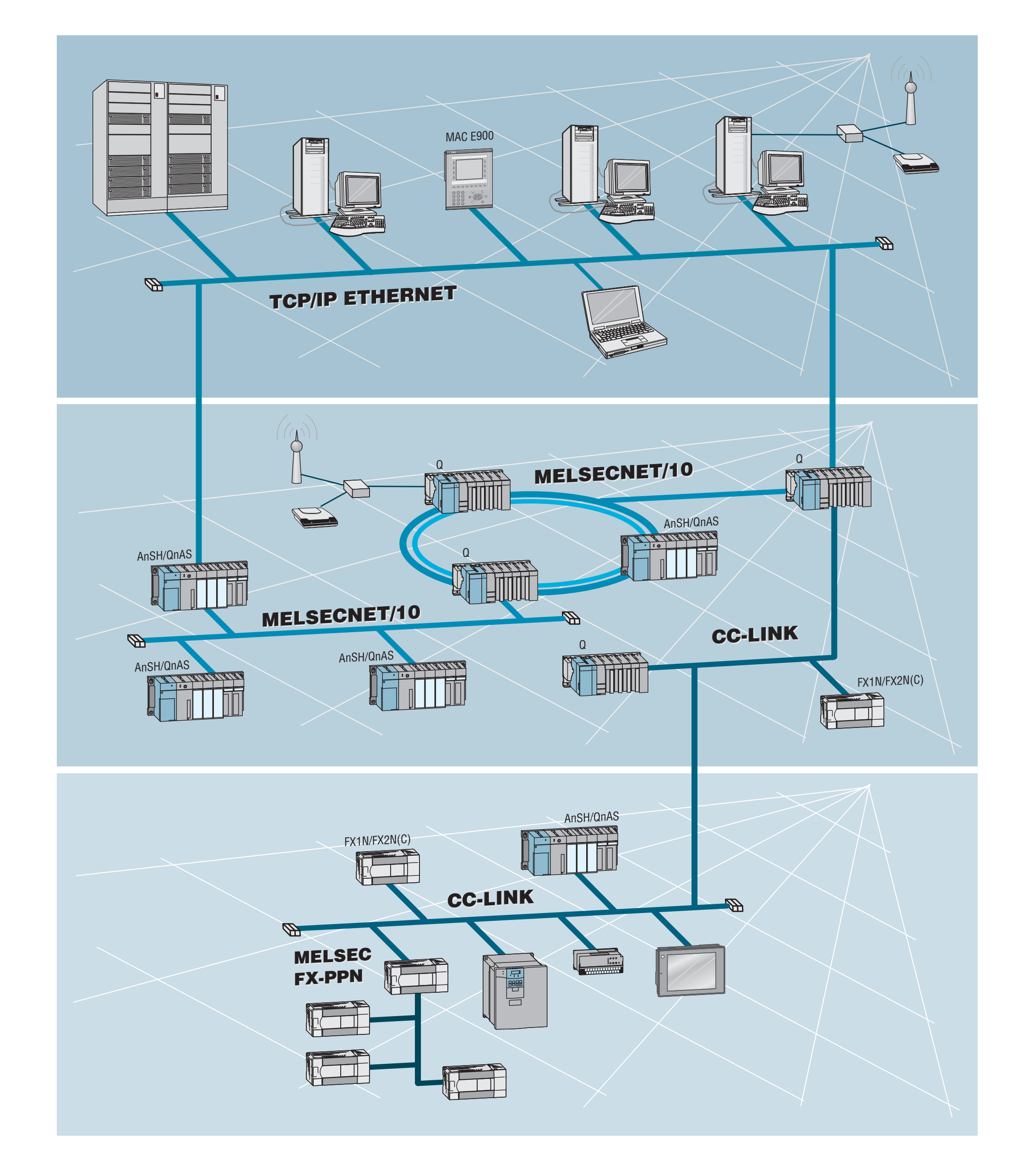
Netzwerkmodule für CANopen stehen für Steuerungen der MELSEC FX-Familie zur Verfügung.

## 3.7.3 MELSEC-Netzwerke

**Leitebene**
TCP/IP ETHERNET

**Steuerungsebene**
CC-Link
MELSECNET/10
MELSECNET/H

**Fertigungsebene**
CC-Link
MELSEC FX-PPN




### MELSECNET/10/H

Über das MELSECNET/10 und MELSECNET/H wird ein äußerst schneller zyklischer Datenaustausch zwischen MELSEC SPS realisiert. Selbst dezentrale E/A-Baugruppen lassen sich auf diese Weise vernetzen. Über das MELSECNET/10/H lässt sich jede SPS von einer beliebigen Station aus programmieren und überwachen.

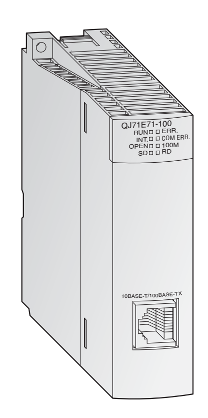
Insgesamt können bis zu 255 MELSECNET/10-Netzwerke miteinander verknüpft werden. Aufgrund der eingebauten Router-Funktion lassen sich Daten sehr leicht von einem Netzwerk in ein anderes Netzwerk übergeben. Für den zyklischen Datenaustausch steht eine äußerst große Datenmenge von 8192 Worten und 8192 Merkern zur Verfügung. Als Besonderheit können parallel zum zyklischen Datenaustausch von einer beliebigen Station – auch über mehrere Netzwerke hinweg – Daten zu einer beliebigen anderen Station gesendet oder von dieser Station gelesen werden.

Für den Datenaustausch stehen verschiedenste Kabeltypen und Topologien zur Verfügung, vom koaxialen Bus (max. 500 m) über einen koaxialer Doppelring bis zum optischer Doppelring für Netzwerkausdehnungen von bis zu 30 km(!).

## 3.7.4 Netzwerk-Module

### ETHERNET-Module

Mit den Modulen QJ71E71 und QD71E71-B2 kann das MELSEC System Q über das ETHERNET mit anderen Geräten, wie z. B. einem Personal Computer, verbunden werden. Neben dem Datenaustausch per TCP/IP- oder UDP/IP-Kommunikation können über das ETHERNET auch SPS-Daten gelesen oder geändert sowie der Betrieb und der Zustand der CPU überwacht werden.

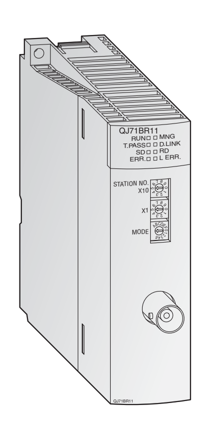



**Besondere Merkmale:**

- 10BASE5-, 10BASE2- oder 10BASE-T-Schnittstelle
- Übertragungsgeschwindigkeit von 10 bzw. 100 Mbit/s
- FTP-Serverfunktion ist möglich
- Datenaustausch über Sende- und Empfangspuffer mit fester Größe
- Bis zu 16 logische Verbindungen können gleichzeitig aufgebaut werden.
- Mit einem PC, auf dem die Software GX Developer oder GX IEC Developer installiert ist, kann das Programm der SPS über das ETHERNET geändert werden.

### MELSECNET-Module

Die Module QJ71BR11 und QJ71LP21 ermöglichen die Anbindung des MELSEC System Q an ein MELSECNET/10 oder MELSECNET/H und damit die Kommunikation mit Steuerungen der Q-, QnA und QnAS-Serie.




**Besondere Merkmale:**

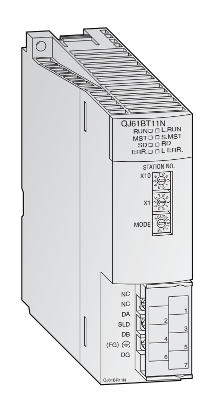
- Zwei verschiedene Netzwerk-Topologien können genutzt werden: Koaxialer Bus (QJ71BR11) oder optischer Doppelring (QJ71LP21)
- Hohe Übertragungsgeschwindigkeit: 10 Mbit/s bei koaxialem Bus und wahlweise 10 oder 20 Mbit/s bei optischem Doppelring
- Datenaustausch mit SPS/PC und dezentralen E/A-Stationen ist möglich
- Daten können mit beliebigen Stationen ausgetauscht werden, unabhängig davon, wie viele Netzwerke zwischen den Stationen liegen.
- Ausblendung einer fehlerhaften Station beim koaxialen Bus und Loopback-Funktion beim optischem Doppelring, wenn eine Station gestört ist.
- Bei Ausfall der Kontrollstation übernimmt eine andere Station automatisch deren Aufgabe

### Master-Modul/Lokales Modul für CC-Link

Das QJ61BT11N ist in einem CC-Link-System als Master- oder lokale Station einsetzbar und dient zur Steuerung und Überwachung von dezentralen Ein- und Ausgängen.



**Besondere Merkmale:**

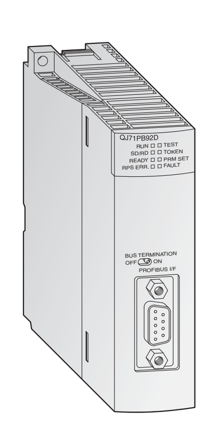



- Die Parametrierung aller im Netzwerk vorhandenen Module erfolgt direkt über das Master-Modul.
- Automatische Kommunikation zwischen den dezentralen Geräten und dem Master-Modul. Die Abtastzeit für 2048 E/As beträgt nur 3,3 ms.
- Übertragungsgeschwindigkeiten von bis zu 10 Mbit/s
- Erweiterung eines Systems um bis zu 2048 dezentrale E/As durch ein Master-Modul
- Mit einem zusätzlichen Stand-by-Master kann ein redundantes System aufgebaut werden. Nach Ausfall der Master-Station wird die Kommunikation fortgesetzt.
- Automatischer Start des CC-Link ohne Parametrierung
- Abhängig von Bedingungen im Netzwerk können Interrupt-Programme gestartet werden.

### PROFIBUS/DP-Modul

Die PROFIBUS/DP-Master-Module QJ71PB92D und QJ71PB92V sowie das PROFIBUS/DP-Slave-Modul QJ71PB93D erlauben den Datenaustausch von Steuerungen im MELSEC System Q mit anderen Geräten an einem PROFIBUS/DP-Netzwerk.

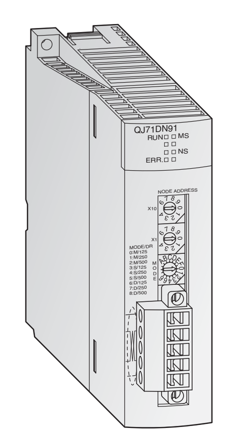
**Besondere Merkmale:**




- Die Master-Station kann mit bis zu 60 Slave-Stationen Daten austauschen.
- Pro Slave können 244 Eingangs- und 244 Ausgangsbytes verarbeitet werden.
- Globale Dienste wie SYNC und FREEZE sowie Diagnosefunktionen für bestimmte Slaves werden unterstützt.
- Der Datenaustausch kann über automatisch und zusätzlich durch Blockanweisungen erfolgen.

### DeviceNet-Master-Modul QJ71DN91

Das QJ71DN91 verbindet eine SPS des MELSEC System Q mit dem DeviceNet. Das DeviceNet ist eine preiswerte Lösung zur Netzwerkanbindung von „Low-Level“-Endgeräten.




**Besondere Merkmale:**

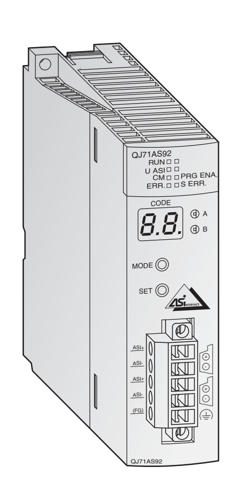
- Die Positionen der Master-Station und der Slave-Stationen können vom Anwender frei gewählt werden.
- Übertragungsgeschwindigkeiten von 125, 250 oder 500 kBit/s
- Die Leitungslänge kann bis 500 m betragen.
- Kommunikationsmethoden:
  - Polling
  - Bit strobe
  - Zustandsänderung
  - Zyklisch

### Master-Module für AS-Interface

Für die Verbindung zwischen den SPS des MELSEC System Q mit dem AS-Interface dient das QJ71AS92 als Master-Modul.

Das QJ71AS92 kann bis zu 62 Slave-Einheiten (31 Gruppe A, 31 Gruppe B) mit jeweils bis zu 4 Eingängen und 4 Ausgängen pro Adresse ansteuern. Die Adressenzuordnung der Slave-Geräte im AS-Interface erfolgt dabei automatisch über den Master.

Die maximale Übertragungsdistanz beträgt ohne Repeater 100 m. Durch die Verwendung von zwei Repeatern kann die Übertragungsdistanz auf bis zu 300 m vergrößert werden.

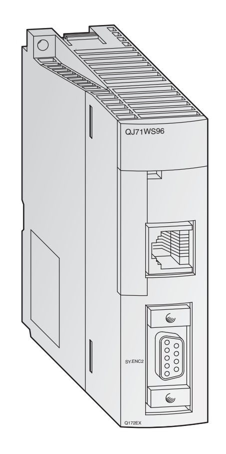



**Besondere Merkmale:**

- Bis zu 62 Slave-Einheiten können in zwei Netzwerken konfiguriert werden.
- Bis zu 496 digitale Ein-/Ausgänge können über den Master betrieben werden.
- Übertragung über eine spezielle 2-Draht-Leitung
- Hochwirksames Fehlersicherungssystem
- Automatischer Austausch der Daten mit der SPS

### Web-Server-Modul

Durch das Web-Server-Modul QJ71WS96 wird die Fernüberwachung einer SPS des MELSEC System Q ermöglicht.




**Besondere Merkmale:**

- Zugriff auf die Steuerung via Internet
- Einfachste Parametrierung
- Nutzer benötigt für Einstellungen und zur Fernüberwachung nur einen Web-Browser.
- RS232-Schnittstelle zum Anschluss eines Modems
- Für die Kommunikation können verschiedene Netzwerkverbindungen genutzt werden: ADSL, Modem, LAN, etc.
- Senden und Empfangen von Daten per E-Mail oder FTP
- Selbstgestaltete Webseiten und Java-Applets integrierbar
- Standard-Verbindung über ETHERNET für den Datenaustausch mit anderen Steuerungen oder PCs
- Erfassung und Speicherung von Ereignissen und CPU-Zuständen

# 4 Grundlagen der Programmierung

Ein Programm besteht aus einer Folge einzelner Steuerungsanweisungen, die die Funktion der Steuerung festlegen und die von der SPS in der programmierten Reihenfolge abgearbeitet werden. Bei der Programmierung muss der eigentliche Steuerungsprozess daher in einzelne Anweisungen zerlegt werden. Eine Steuerungsanweisung ist die kleinste Einheit eines SPS-Anwenderprogramms.

## 4.1 Aufbau einer Steuerungsanweisung

Eine Steuerungsanweisung besteht aus einer Anweisung (Befehl) und einem oder – bei Applikationsanweisungen– auch mehreren Operanden. Einige Steueranweisungen kommen auch ohne Operanden aus. Diese Anweisungen steuern die Programmverarbeitung in der SPS.

Bei der Programmierung wird jede Steuerungsanweisung automatisch mit einer Schrittnummer versehen und dadurch deren Position im Programm eindeutig festgelegt, denn die gleiche Anweisung mit demselben Operanden kann ja auch mehrmals im Programm verwendet werden.

Darstellung einer Anweisung im Kontaktplan (links) und in Anweisungsliste (rechts):




Der Befehl beschreibt, was zu tun ist, also die Funktion, die die Steuerung ausführen soll. Der Operand gibt an, womit etwas zu tun ist. Seine Bezeichnung setzt sich aus dem Operandenkennzeichen und der Operandenadresse zusammen.




Beispiele für Operandenkennzeichen:

| Operandenkennzeichen | Typ | Bedeutung |
|---|---|---|
| X | Eingang | Eingangsklemme der SPS (z. B. Schalter) |
| Y | Ausgang | Ausgangsklemme der SPS (z. B. Schütz oder Lampe) |
| M | Merker | Zwischenspeicher in der SPS, der zwei Zustände („Ein“ oder „Aus“) annehmen kann |
| T | Timer | „Zeitrelais" zur Realisierung von zeitabhängigen Funktionen |
| C | Counter | Zähler |
| D | Datenregister | Datenspeicher in der SPS, in dem z. B. Messwerte oder Rechenergebnisse abgelegt werden können. |

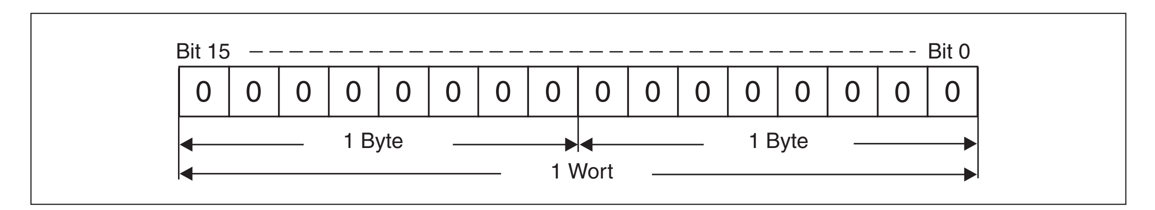
Die Operanden sind im Kapitel 5 ausführlich beschrieben.

Da zum Beispiel mehrere Eingänge vorhanden sind, wird durch die Angabe der Operandenadresse ein individueller Eingang festgelegt.

## 4.2 Bits, Bytes und Worte

Die kleinste Informationseinheit einer SPS (und in der Digitaltechnik allgemein) ist das „Bit“. Ein Bit kann nur zwei Zustände annehmen: „0“ (ausgeschaltet oder unwahr (FALSE)) und „1“ (eingeschaltet oder wahr (TRUE)). Bits begegnen Ihnen in der SPS zum Beispiel in Form von Eingängen, Ausgängen und Merkern, den sogenannten **Bit-Operanden**.

8 Bits ergeben ein Byte, zwei Bytes bilden ein Wort. In einer SPS des MELSEC System Q gehören zum Beispiel die Datenregister zu den **Wort-Operanden**.




Durch ihre Größe von 16 Bit können in jedem Register Werte im Bereich von -32768 bis 32767 (vgl. Abschnitt 4.3) gespeichert werden. Falls das nicht ausreichend ist, können zwei Worte zu einem Doppelwort mit 32 Bits zusammengefasst werden, in dem dann Werte von -2 147 483 648 bis 2 147 483 647 abgelegt werden können.

## 4.3 Zahlensysteme

In einer SPS des MELSEC System Q werden verschiedene Zahlensysteme verwendet. Sie dienen zur Eingabe oder Anzeige von Werten und zur Angabe einer Operandenadresse.

### Dezimale Zahlen

Mit dezimalen Zahlen gehen wir täglich um. Ihre Basis ist „10“, das heißt nach einer Zählung bis 9 erfolgt bei weiterer Zählung jeweils ein Übertrag in die nächste Dekade (9 → 10, 19 → 20, 29 → 30 usw.).

– Basis: 10

– Ziffern: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Mit dezimalen Zahlen werden beim MELSEC System Q Konstante und Sollwerte von Timern und Countern angegeben. Außerdem erfolgt –außer bei Ein- und Ausgängen– die Angabe der Operandenadressen im dezimalen Format.

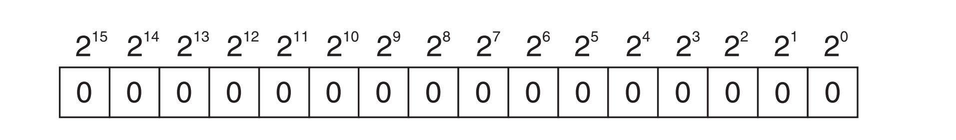
### Binäre Zahlen (Dualzahlensystem)

Eine SPS verarbeitet wie alle Computer nur EIN/AUS- oder 0/1-Informationen, die in einzelnen Bits gespeichert sind (binäre Informationen). Bei der Eingabe oder Anzeige von Zahlen in anderen Formaten wandelt die Programmier-Software die verschiedenen Zahlensysteme automatisch um.

– Basis: 2

– Ziffern: 0 und 1

Wenn binäre Zahlen in einem Wort gespeichert werden, erhalten die einzelnen Bits bestimmte Wertigkeiten:




| Darstellung mit Basis 2 | Dezimaler Wert | Darstellung mit Basis 2 | Dezimaler Wert |
|---|---|---|---|
| $2^0$ | 1 | $2^8$ | 256 |
| $2^1$ | 2 | $2^9$ | 512 |
| $2^2$ | 4 | $2^{10}$ | 1024 |
| $2^3$ | 8 | $2^{11}$ | 2048 |
| $2^4$ | 16 | $2^{12}$ | 4096 |
| $2^5$ | 32 | $2^{13}$ | 8192 |
| $2^6$ | 64 | $2^{14}$ | 16384 |
| $2^7$ | 128 | $2^{15}$ | 32768* |

* Bit 15 wird bei binären Werten zur Kennzeichnung des Vorzeichens verwendet. (Bit 15 = 0: Positiver Wert, Bit 15 = 1: Negativer Wert)

Zur Umwandlung einer binären Zahl in eine dezimale Zahl werden die Bits, die „1“ sind, entsprechend ihrer Wertigkeit in einen dezimalen Wert gewandelt und anschließend die einzelnen Werte addiert.

**Beispiel** ▽

00000010 00011001 (binär)

00000010 00011001 (binär) = $1 \times 2^9 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^0$
00000010 00011001 (binär) = 512 + 16 + 8 + 1
00000010 00011001 (binär) = 537 (dezimal)

**Hexadezimalzahlensystem**

Hexadezimale Zahlen lassen sich leicht aus binären Zahlen erzeugen und werden aus diesem Grund oft in der Digitaltechnik und bei speicherprogrammierbaren Steuerungen verwendet. Bei Steuerungen des MELSEC System Q werden mit hexadezimalen Zahlen die Adressen der Ein- und Ausgänge sowie Konstante angegeben. In der Programmieranleitung und in den Handbüchern zu den Modulen sind hexadezimale Zahlen immer durch ein angefügtes „H“ gekennzeichnet, um eine Verwechslung mit dezimalen Zahlen zu vermeiden (z. B. 12345H)

– Basis: 16

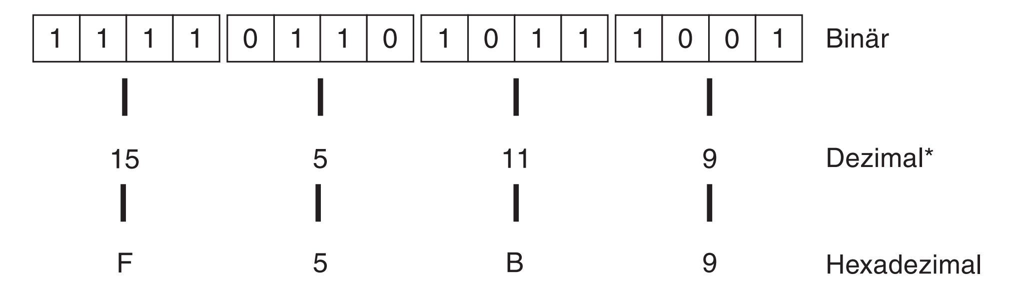
– Ziffern: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
(Die Buchstaben A, B, C, D, E und F entsprechen den dezimalen Werten 10 bis 15.)

Im hexadezimalen System erfolgt nach einer Zählung bis FH bei weiterer Zählung jeweils ein Übertrag in die nächste Stelle (FH → 10H, 1FH → 20H, 2FH → 30H). Jede Stelle hat eine Wertigkeit mit der Basis 16.

1A7FH

$16^0 = 1$ (In diesem Beispiel: 15 x 1 = 15)
$16^1 = 16$ (In diesem Beispiel: 7 x 16 = 112)
$16^2 = 256$ (In diesem Beispiel: 10 x 256 = 2560)
$16^3 = 4096$ (In diesem Beispiel: 1 x 4096 = 4096)
6783 (Dezimal)

Die oben erwähnte einfache Umwandlung binärer in hexadezimaler Zahlen und umgekehrt soll am folgenden Beispiel demonstriert werden:




* Bei der Umwandlung in dezimale Werte werden jeweils 4 Bits umgewandelt. Die so entstandene Dezimalzahl entspricht nicht dem Wert der kompletten 16-Bit-Binärzahl!

### Oktalzahlensystem

Das oktale Zahlensystem ist hier nur der Vollständigkeit halber aufgeführt. Bei einer SPS des MELSEC System Q wird es nicht verwendet. Beim oktalen System wird als Basis die „8“ verwendet, daher existieren die Zahlen 8 und 9 nicht. Nach einer Zählung bis 7 erfolgt bei weiterer Zählung jeweils ein Übertrag in die nächste Stelle (0 bis 7, 10 bis 17 .... 70 bis 77, 100 bis 107 usw.).

– Basis: 8
– Ziffern: 0, 1, 2, 3, 4, 5, 6, 7

### Zusammenfassung

In der folgenden Tabelle sind die oben beschriebenen vier Zahlensysteme noch einmal gegenübergestellt:

| Dezimalzahl | Oktalzahl | Hexadezimalzahl | Binärzahl |
|---|---|---|---|
| 0 | 0 | 0 | 0000 0000 0000 0000 |
| 1 | 1 | 1 | 0000 0000 0000 0001 |
| 2 | 2 | 2 | 0000 0000 0000 0010 |
| 3 | 3 | 3 | 0000 0000 0000 0011 |
| 4 | 4 | 4 | 0000 0000 0000 0100 |
| 5 | 5 | 5 | 0000 0000 0000 0101 |
| 6 | 6 | 6 | 0000 0000 0000 0110 |
| 7 | 7 | 7 | 0000 0000 0000 0111 |
| 8 | 10 | 8 | 0000 0000 0000 1000 |
| 9 | 11 | 9 | 0000 0000 0000 1001 |
| 10 | 12 | A | 0000 0000 0000 1010 |
| 11 | 13 | B | 0000 0000 0000 1011 |
| 12 | 14 | C | 0000 0000 0000 1100 |
| 13 | 15 | D | 0000 0000 0000 1101 |
| 14 | 16 | E | 0000 0000 0000 1110 |
| 15 | 17 | F | 0000 0000 0000 1111 |
| 16 | 20 | 10 | 0000 0000 0001 0000 |
| : | : | : | : |
| 99 | 143 | 63 | 0000 0000 0110 0011 |
| : | : | : | : |

# 4.4 Codes

Um Informationen schnell und sicher zu übertragen, werden z. B. die Buchstaben des Alphabets und die Dezimalziffern in eine maschinengerechte Sprache übersetzt (codiert).

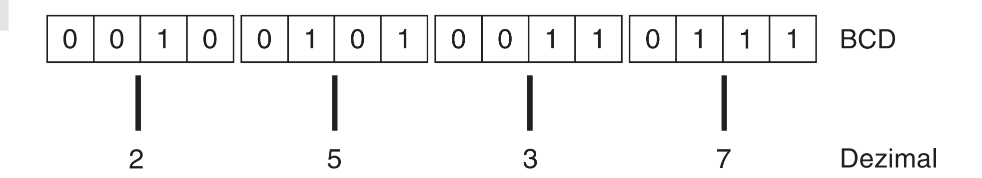
## 4.4.1 BCD-Code

BCD oder BCD-Code (von *Binary Coded Decimal* = dualcodierte Dezimalziffer), ist eine Codierung, bei der jede dezimale Ziffer (0 bis 9) durch jeweils 4 Bit im Dualsystem (0000 bis 1001, siehe Tabelle) dargestellt wird. Mit einem Byte (8 Bit) können also zwei Dezimalziffern dargestellt werden.

| Dezimalzahl | BCD-Code |
|---|---|
| 0 | 0000 |
| 1 | 0001 |
| 2 | 0010 |
| 3 | 0011 |
| 4 | 0100 |
| 5 | 0101 |
| 6 | 0110 |
| 7 | 0111 |
| 8 | 1000 |
| 9 | 1001 |

Zur Codierung von Zahlen mit mehr als einer Dezimalziffer werden die BCD-Darstellungen der einzelnen Ziffern hintereinander gesetzt . Eine vierstellige Zahl im BCD-Code belegt ein Wort (16 Bit) und kann die Ziffern 0000 bis 9999 enthalten.

**Beispiel** ▽




Der BCD-Code wird im MELSEC System Q nicht für interne Operationen verwendet. In der Anlagenautomatisierung werden aber zur Eingabe von Zahlenwerten oft Schalter eingesetzt, die BCD-codierte Werte ausgeben oder 7-Segmentanzeigen, denen die Anzeigewerte im BCD-Code übergeben werden müssen. Für diese Fälle stehen verschiedene Anweisungen zur Konvertierung aus und in den BCD-Code zur Verfügung.

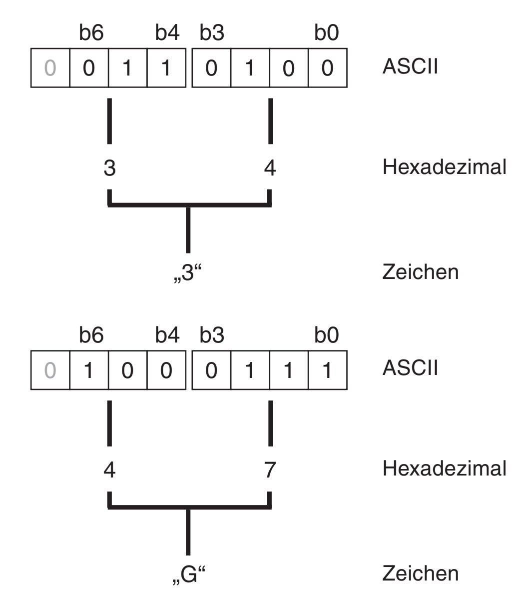
## 4.4.2 ASCII-Code

ASCII ist die Abkürzung für ***A**merican **S**tandard **C**ode for **I**nformation **I**nterchange* (amerikanischer Standard-Code für den Datenaustausch). Beim ASCII-Code lassen sich mit 7 Bits neben alphanumerischen Zeichen auch Sonderzeichen und Steuerbefehle darstellen.

Daten im ASCII-Code werden zum Datenaustausch mit Peripheriegeräten verwendet.

| Bits 3 bis 0 | | Bits 6 bis 4 | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| | | **000** | **001** | **010** | **011** | **100** | **101** | **110** | **111** |
| **0** | **0000** | NUL | DLE | SP | 0 | @ | P | ′ | p |
| **1** | **0001** | SOH | DC1 | ! | 1 | A | Q | a | q |
| **2** | **0010** | STX | DC2 | !! | 2 | B | R | b | r |
| **3** | **0011** | ETX | DC3 | # | 3 | C | S | c | s |
| **4** | **0100** | EOT | DC4 | $ | 4 | D | T | d | t |
| **5** | **0101** | ENQ | NAK | % | 5 | E | U | e | u |
| **6** | **0110** | ACK | SYN | & | 6 | F | V | f | v |
| **7** | **0111** | BEL | ETB | ‘ | 7 | G | W | g | w |
| **8** | **1000** | BS | CAN | ( | 8 | H | X | h | x |
| **9** | **1001** | HT | EM | ) | 9 | I | Y | i | y |
| **A** | **1010** | LF | SUB | * | : | J | Z | j | z |
| **B** | **1011** | VT | ESC | + | ; | K | [ | k | { |
| **C** | **1100** | FF | FS | , | < | L | \ | l | \| |
| **D** | **1101** | CR | GS | - | = | M | ] | m | } |
| **E** | **1110** | SO | RS | . | > | N | ↑ | n | ~ |
| **F** | **1111** | SI | VS | / | ? | O | ← | o | DEL |

**Beispiele** ▽



# 4.5 Programmiersprachen

Die Software GX IEC Developer bietet verschiedene Möglichkeiten zur Programmierung. Es kann zwischen grafischer Eingabe und Anzeige des Programms und Eingabe/Anzeige in Textform gewählt werden. Mit Ausnahme der Ablaufsprache kann ein Programm bei allen Programmiersprachen in einzelne Abschnitte, sogenannten Netzwerken, unterteilt werden.

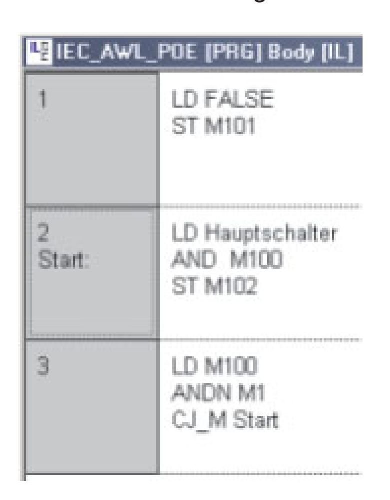
## 4.5.1 Text-Editoren

### Anweisungsliste (AWL)

Bei der Programmierung in Form einer Anweisungsliste werden die Steuerungsanweisungen als Text eingegeben. Eine Anweisungsliste besteht aus einer Folge von Steuerungsanweisungen. Jede Steuerungsanweisung muss in einer separaten Zeile eingetragen werden.

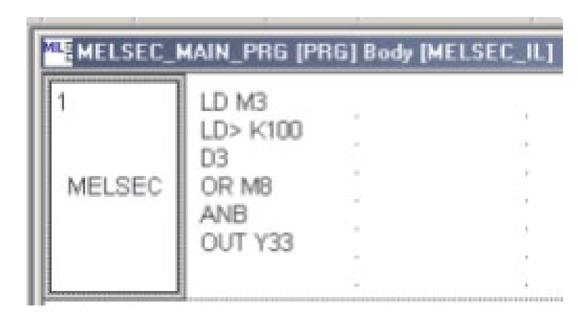
Es werden zwei verschieden Arten von Anweisungslisten verwendet:

- IEC-Anweisungsliste




- MELSEC-Anweisungsliste

In einer MELSEC-Anweisungsliste können nur MELSEC-Anweisungen verwendet werden, eine Programmierung nach dem IEC-Standard ist nicht möglich.




### Strukturierter Text (ST)

Strukturierter Text ist ein nützliches Werkzeug. Besonders Programmierer, die mit Hochsprachen vertraut sind, werden dieses Werkzeug zu schätzen wissen. Wenn bei der Programmierung die Wirkungsweise einer SPS berücksichtigt und das Programm sorgfältig vorbereitet wird, ist die Programmierung mit Strukturiertem Text sehr komfortabel.

Der Editor für strukturierten Text ist kompatibel zur IEC 61131-3, es werden alle Anforderungen erfüllt. Die folgende Abbildung zeigt ein Beispiel für die Programmierung mit Strukturiertem Text.

```
(*Beispiel für Strukturierten Text*)

Y20:=X10;
Y21:=X11 AND X12 OR X13;
M0:=(M1 AND (M2 OR M3)) OR X14;
```

## 4.5.2 Grafik-Editoren

### Kontaktplan (KOP)

Die Programmierung im Kontaktplan ist ähnlich dem Zeichen von Schaltplänen für konventionelle Schützsteuerungen. Ein Kontaktplan besteht aus Eingangskontakten (Schließer und Öffner), Ausgängen (Spulen), aber auch Funktionen und Funktionsbausteinen. Diese Elemente werden durch horizontale Linien verbunden. Untereinander angeordnete Elemente können durch vertikale Linien verbunden werden. So entsteht grafisch das Programm. Ein Netzwerk beginnt im Kontaktplan immer an der linken Sammelschiene.

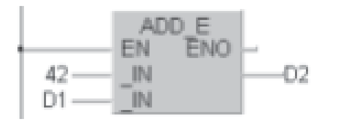


*Beispiel für Kontaktplan*

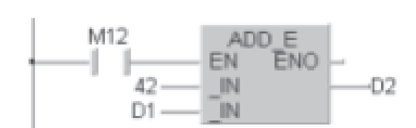
Die am häufigsten benötigten Grundanweisungen können bei der Programmierung im Kontaktplan über die Werkzeugleiste aufgerufen werden.



Komplexere Funktionen und Funktionsbausteine werden in einem Kontaktplan-Programm als Kästen dargestellt. Zusätzlich zu den Ein- und Ausgängen, die für die Funktion erforderlich sind, besitzen Funktionen und Funktionsbausteinen einen EN-Eingang und einen ENO-Ausgang. Über den EN-Eingang (ENable = Freigabe) kann die Ausführung der Funktion oder des Funktionsbausteins gesteuert werden.



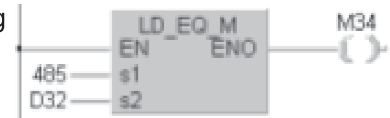

Diese Anweisung wird zyklisch ausgeführt.




Diese Anweisung wird nur ausgeführt, wenn M12 eingeschaltet ist.

Am ENO-Ausgang (ENO = ENable Out) wird das Verknüpfungsergebnis ausgegeben.

Der Merker M34 wird „1“, wenn die Vergleichsbedingung erfüllt ist.

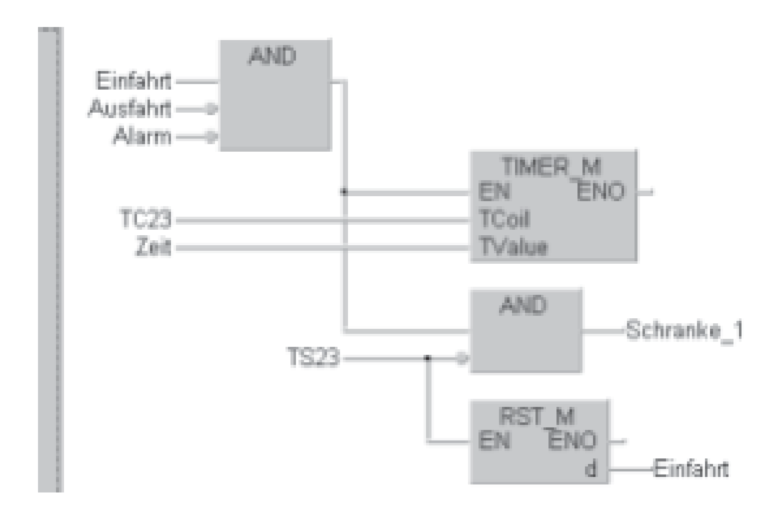



Zur Steuerung des Programmausführung können ENO-Ausgänge und EN-Eingänge verbunden werden. Im folgenden Beispiel ist die Ausführung der zweiten Anweisung vom Ergebnis der ersten Anweisung abhängig.

### Funktionsbausteinsprache (FBS)

Bei der Funktionsbausteinsprache werden alle Elemente als Blöcke angegeben. Diese Blöcke werden mit horizontalen und vertikalen Linien verbunden. Eine Sammelschiene ist nicht vorhanden.

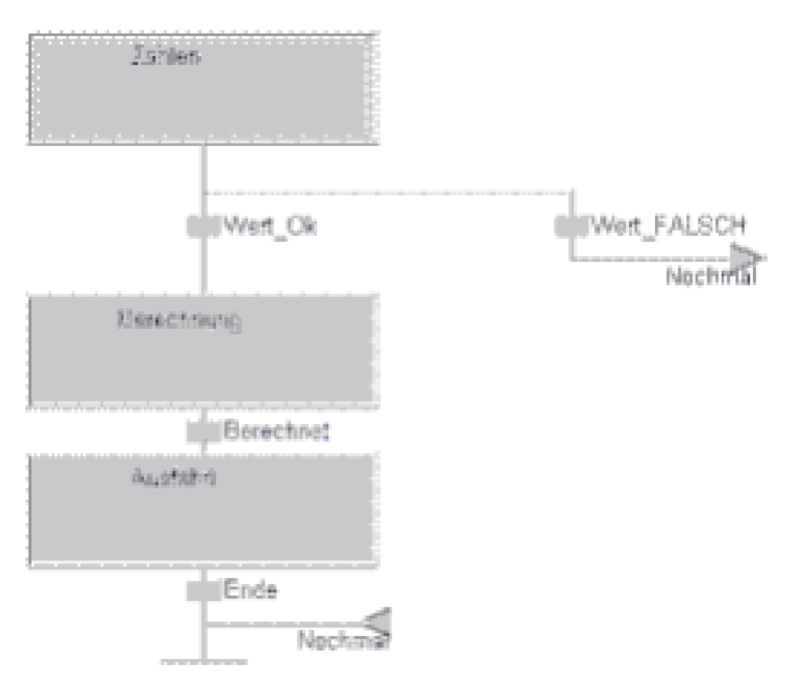
Beispiel für die Programmierung in der Funktionsbausteinsprache:




### Ablaufsprache (AS)

Die Ablaufsprache ist eine grafische Strukturierungssprache, mit der Prozessabläufe übersichtlich dargestellt werden können.

Die Ablaufsprache besteht im wesentlichen aus zwei Grundelementen: Schritte und Transitionen. Ein Prozess besteht aus einer Reihe von Schritten, die durch Transitionen voneinander getrennt sind. Zur selben Zeit kann nur ein Schritt des Programms aktiviert sein. Ein Schritt wird nur aktiviert, wenn der vorherige Schritt vollständig abgearbeitet wurde und die Bedingung für die Transition erfüllt ist.



# 4.6 Programmierung nach der Norm IEC 61131-3

Die Norm IEC 61131-3 ist der internationale Standard für SPS-Programme. (IEC: *International Electromechanical Commission*). IEC 61131-3 umfasst nicht nur die SPS-Programmiersprachen, sondern bietet auch umfassende Konzepte und Richtlinien zum Aufbau eines SPS-Projekts.

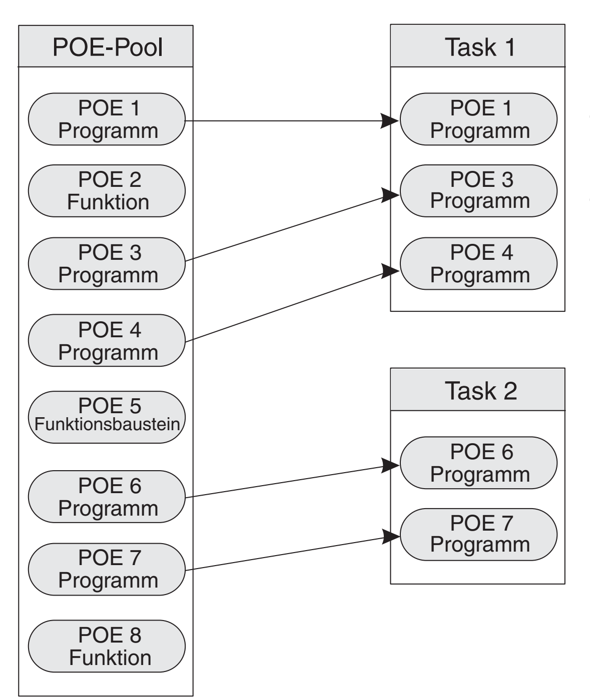
Mit der Programmier-Software GX IEC Developer können speicherprogrammierbare Steuerungen der Norm IEC 61131-3 entsprechend programmiert werden.

In diesem Einsteigerhandbuch werden nur die Begriffe erklärt, die zum Verständnis der Beispiele notwendig sind. Weitere Informationen zum Umgang mit dem GX IEC Developer finden Sie im Einsteigerhandbuch (Art.-Nr. 43594) und im Benutzerhandbuch zum GX IEC Developer (Art.-Nr. 43595). Während der Programmierung können Sie auch die Hilfe-Funktion des GX IEC Developer nutzen.

## 4.6.1 Struktur der Programme

**Programm-Organisationseinheiten (POE)**

Ein Gesamtprogramm wird in der IEC 61131-3 in Teilprogramme, den Programm-Organisationseinheiten (POE) aufgeteilt. Eine POE ist die kleinste unabhängige Software-Einheit eines Ablauf-Programmes.




Die POEs werden im POE-Pool abgelegt.

Die Programm-POEs werden zu einer Gruppe (Task) zusammengefasst.

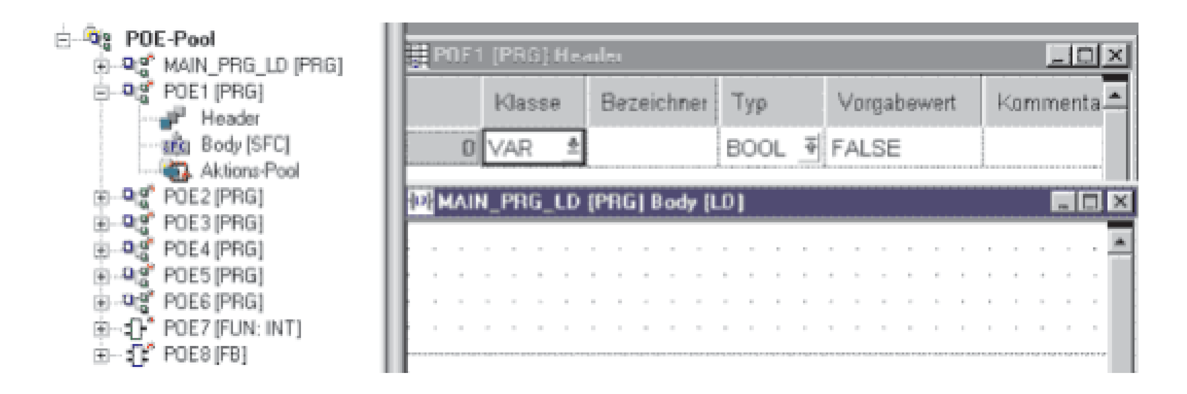
Die einzelnen Task wiederum bilden das Gesamtprogramm.

Jede Programm-Organisationseinheit setzt sich zusammen aus:

- dem Header und
- dem Body

Im ***Header*** werden die Variablen festgelegt, die in dieser POE verwendet werden.

Der ***Body*** enthält das eigentliche SPS-Programm in den verschiedenen Sprachen wie z. B. Kontaktplan oder IEC-Anweisungsliste.

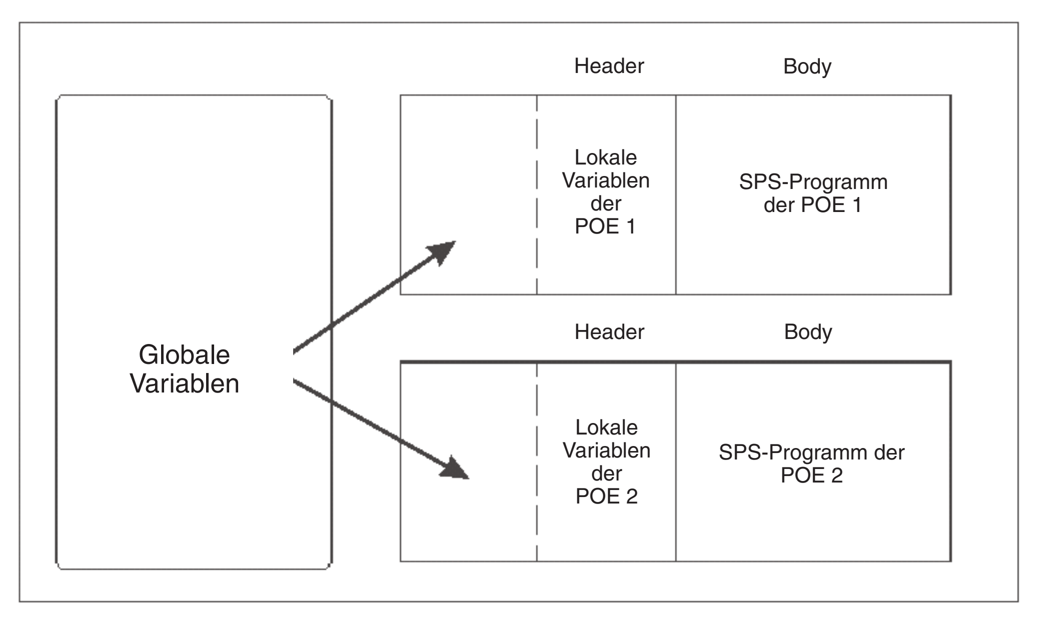



## 4.6.2 Die Variablen

Variablen enthalten die Werte von Eingängen, Ausgängen oder internen Operanden der SPS. Man unterscheidet zwischen

- Globalen Variablen und
- Lokalen Variablen

**Globale Variablen** können als gemeinsame Variablen angesehen werden. Sie sind die Schnittstelle zu den physischen SPS-Operanden wie zum Beispiel Ein- oder Ausgängen. Globale Variablen gelten für das gesamte Programm und können in allen POEs verwendet werden. Sie verweisen entweder auf vorhandene Ein- und Ausgänge der SPS oder auf interne SPS-Operanden. Globale Variablen ermöglichen einen Datenaustausch zwischen den POEs




Damit in einer POE auf eine Globale Variable zugegriffen werden kann, muss diese Variable im Header der POE eingetragen werden. Im Header können Lokale und Globale Variablen eingetragen werden.

Eine **Lokale Variable** kann als Speicher für ein Zwischenergebnis angesehen werden. Auf diese Variablen können andere POEs nicht zugreifen.

### Deklaration der Variablen

Zu Beginn jeder POE werden die Variablen deklariert, d.h. sie werden einem bestimmten Datentyp (wie INT oder BOOL) zugeordnet.

Jede Variable wird gekennzeichnet durch:

- das Schlüsselwort
- den Bezeichner, den Namen der Variablen
- die absolute Adresse (optional für Globale Variablen)
- den Datentyp
- den Anfangswert (wird automatisch vergeben)
- den Kommentar (bei Bedarf)




### Schlüsselwörter

Mit dem Schlüsselwort werden der Variablen bestimmte Eigenschaften zugeordnet,die die Verwendung der Variablen im Projekt kennzeichnen. Einige Beispiele:

– VAR: Lokale Variable innerhalb der POE

– VAR_EXTERNAL: Externe Variable, die in der Globalen Variablenliste deklariert wird und für alle POEs les- und beschreibbar ist.

– VAR_CONSTANT: Lokale Variable mit festen Wert

### Bezeichner

Jede Variablen erhält eine symbolische Adresse. Dieser individuelle Name (Bezeichner) kann frei gewählt werden, muss aber mit einem Buchstaben oder einem (einzelnen) Unterstrich beginnen.

Beispiele für Bezeichner:

– S02.3

– Antrieb_2_bereit

– Ventil_oeffnen

– Motor_M1_EIN

Die Verwendung symbolischer Namen entspricht der Norm IEC 61131-3.

### Absolute Adressen

Globalen Variablen sollten absolute Adressen zugeordnet werden, da diese sonst automatisch vergeben werden. Eine absolute Adresse kennzeichnet einen bestimmten Speicherplatz der Variablen in der CPU oder einen Ein- oder Ausgang.

Die absolute Adresse kann sowohl mit der IEC-Syntax (IEC-Adresse) als auch mit der MITSUBISHI-Syntax (MELSEC-Syntax) eingegeben werden. Beispiele für absolute Adressen:

Eingang X0F = X0F (MELSEC-Syntax) = %IX15 (IEC-Adresse)

Ausgang Y03 = Y03 (MELSEC-Syntax) = %QX3 (IEC-Adresse)

### Elementare Datentypen

Durch den Datentyp werden die Eigenschaften einer Variablen, wie Wertebereich und Anzahl der Bits, festgelegt.

| **Datentyp** | | **Wertebereich** | **Größe** |
|---|---|---|---|
| BOOL | Boolean | 0 (FALSE), 1 (TRUE) | 1 Bit |
| INT | Ganzzahl | -32768 bis +32767 | 16 Bit |
| DINT | Doppelte Ganzzahl | -2.147.483.648 bis 2.147.483.647 | 32 Bit |
| WORD | Bitfolge 16 | 0 bis 65535 | 16 Bit |
| DWORD | Bitfolge 32 | 0 bis 4.294.967.295 | 32 Bit |
| REAL | Gleitkommazahl | 3,4E +/-38 (7 Stellen) | |
| TIME | Zeitwert | -T#24d0h31m23s64800ms bis T#24d20h31m23s64700ms | |
| STRING | Zeichenfolge | Stringkonstanten dürfen nicht länger als 16 Zeichen sein | |

# 4.7 Grundbefehlssatz

Die Anweisungen einer SPS des MELSEC System Q können in einen Grundbefehlssatz und den sogenannten Applikationsanweisungen unterteilt werden.

Die Funktionen der Anweisungen des Grundbefehlssatzes sind mit denen vergleichbar, die in herkömmlichen Schaltungen durch die Verdrahtung entstehen.

### Der Grundbefehlssatz in der Übersicht

| Anweisung | Bedeutung | Beschreibung | Referenz |
|---|---|---|---|
| **LD** | Lade | Beginn einer Verknüpfung mit Abfrage auf Signalzustand „1“ | Abschnitt 4.7.1 |
| **LDI** | Lade invers | Beginn einer Verknüpfung mit Abfrage auf Signalzustand „0“ | |
| **OUT** | Ausgabeanweisung | Zuweisung eines Verknüpfungsergebnisses | Abschnitt 4.7.2 |
| **AND** | UND | UND-Verknüpfung mit Abfrage auf Signalzustand „1“ | Abschnitt 4.7.4 |
| **ANI** | UND-Nicht | UND-Verknüpfung mit Abfrage auf Signalzustand „0“ | |
| **OR** | ODER | ODER-Verknüpfung mit Abfrage auf Signalzustand „1“ | Abschnitt 4.7.5 |
| **ORI** | ODER-Nicht | ODER-Verknüpfung mit Abfrage auf Signalzustand „0“ | |
| **ANB** | UND-Block | Reihenschaltung von parallelen Verknüpfungen | Abschnitt 4.7.6 |
| **ORB** | ODER-Block | Parallelschaltung von in Reihe geschalteten Verknüpfungen) | |
| **LDP** | Flankengesteuerte Verknüpfungen | Lade-Anweisung bei steigender Flanke des Operanden | Abschnitt 4.7.7 |
| **LDF** | | Lade-Anweisung bei fallender Flanke des Operanden | |
| **ANDP** | | UND-Verknüpfung bei steigender Flanke des Operanden | |
| **ANDF** | | UND-Verknüpfung bei fallender Flanke des Operanden | |
| **ORP** | | ODER-Verknüpfung bei steigender Flanke des Operanden | |
| **ORF** | | ODER-Verknüpfung bei fallender Flanke des Operanden | |
| **SET** | Operand setzen | Zuweisung eines Signalzustandes, der auch erhalten bleibt, nachdem die Eingangsbedingung nicht mehr erfüllt ist. | Abschnitt 4.7.8 |
| **RST** | Operand zurücksetzen | | |
| **PLS** | Impuls erzeugen | Setzen eines Operanden für die Dauer eines Programmzyklus bei steigender Flanke der Eingangsbedingung | Abschnitt 4.7.9 |
| **PLF** | | Setzen eines Operanden für die Dauer eines Programmzyklus bei fallender Flanke der Eingangsbedingung | |
| **INV** | Inversion | Verknüpfungsergebnis umkehren | Abschnitt 4.7.10 |
| **FF** | Invertierung eines Bits | Umkehr des Schaltzustandes eines Bit-Ausgangsoperanden | Abschnitt 4.7.11 |
| **MEP** | Impuls aus Verknüpfungsergebnis erzeugen | Pulserzeugung bei steigender Flanke des Verknüpfungsergebnisses | Abschnitt 4.7.12 |
| **MEF** | | Pulserzeugung bei fallender Flanke des Verknüpfungsergebnisses | |

### 4.7.1 Beginn von Verknüpfungen

| Anweisung | Bedeutung | Kontaktplan | IEC-Anweisungsliste |
|---|---|---|---|
| **LD** | Lade-Anweisung, Beginn einer Verknüpfung mit Abfrage auf Signalzustand „1“ | | LD |
| **LDI** | Lade-Anweisung, Beginn einer Verknüpfung mit Abfrage auf Signalzustand „0“ | | LDN |

Ein Strompfad beginnt immer mit einer LD- oder LDI-Anweisung. Als Operanden können Eingänge, Merker, Zeitgeber, aber auch Zähler angegeben werden.

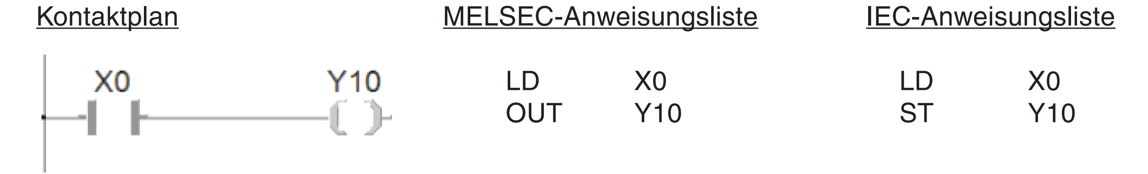
Beispiele zur Anwendung dieser Anweisungen finden Sie im folgenden Abschnitt im Zusammenhang mit der OUT-Anweisung.

### 4.7.2 Ausgabe oder Zuweisung eines Verknüpfungsergebnisses

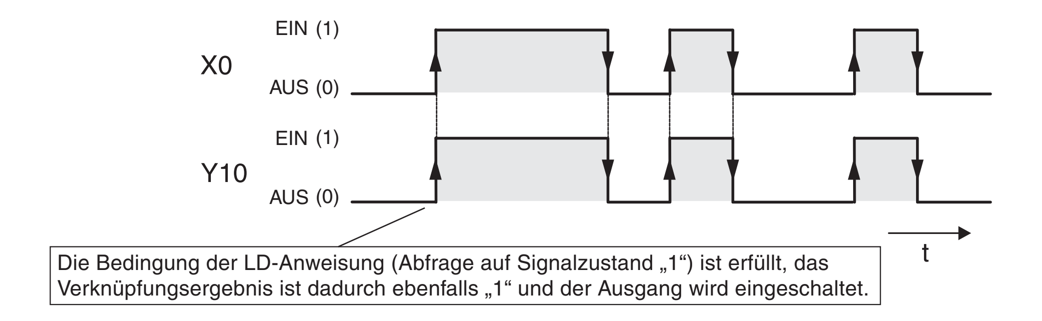
| Anweisung | Bedeutung | Kontaktplan | IEC-Anweisungsliste |
|---|---|---|---|
| **OUT** | Ausgabeanweisung, Zuweisung eines Verknüpfungsergebnisses | | ST |

Mit einer OUT-Anweisung kann ein Strompfad abgeschlossen werden. Es können auch mehrere OUT-Anweisungen als Ergebnis einer Verknüpfung programmiert werden. Das Verknüpfungsergebnis, das einem Operanden mit einer OUT-Anweisung zugewiesen wurde, kann in den nachfolgenden Programmschritten als Eingangssignalzustand verwendet werden.

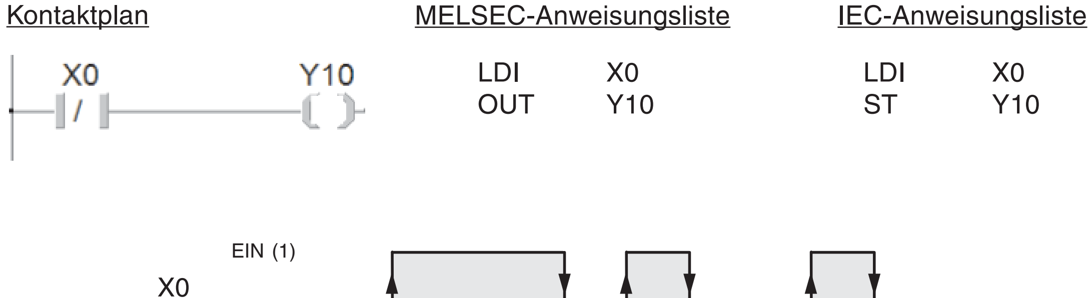
**Beispiel (LD- und OUT-Anweisung)**

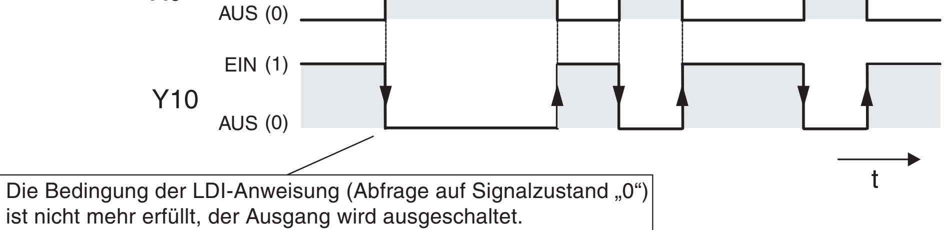



Mit diesen beiden Anweisungen ergibt sich der folgende Signalverlauf:



### Beispiel (LDI- und OUT-Anweisung)



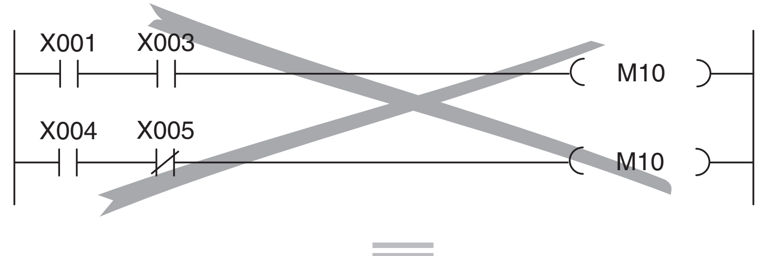



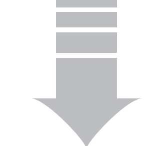

**HINWEIS**

### Doppelbelegung von Merkern oder Ausgängen

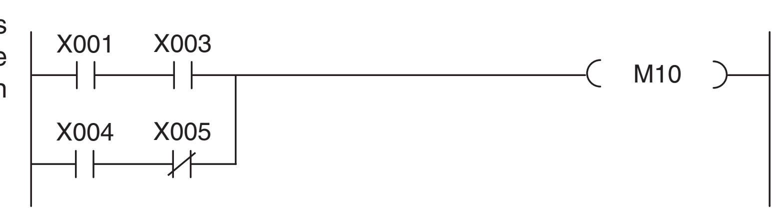
Einem Operanden sollte nur an einer Stelle im Programm ein Verknüpfungsergebnis zugewiesen werden.

Durch die Abarbeitung des Programms von „oben nach unten“ wird die erste Zuweisung für M10 von der zweiten Zuweisung überschrieben.



Durch Modifikation dieses Programmteils werden alle Eingangsverknüpfungen berücksichtigt.




Doch wie bei fast allen Regeln gibt es auch hier Ausnahmen! Sie können sich die sequentielle Bearbeitung des SPS-Programms zu Nutze machen und Anweisungen mit hoher Priorität an das Ende des Programms setzen, um vorherige Verknüpfung absichtlich zu überschreiben. Ein Beispiel dafür finden Sie in Abschnitt 4.9.1. Hier werden Sicherheitseinrichtungen dazu verwendet, interne Operanden der SPS zurückzusetzen und einen Motor anzuhalten. Den Ausgängen für den Motor wird aber nur einmal im gesamten Programm ein Verknüpfungsergebnis zugewiesen!

### 4.7.3 Berücksichtigung der Geber

Bevor weitere Anweisungen beschrieben werden, soll kurz auf die Bedeutung der Gebersignale eingegangen werden.

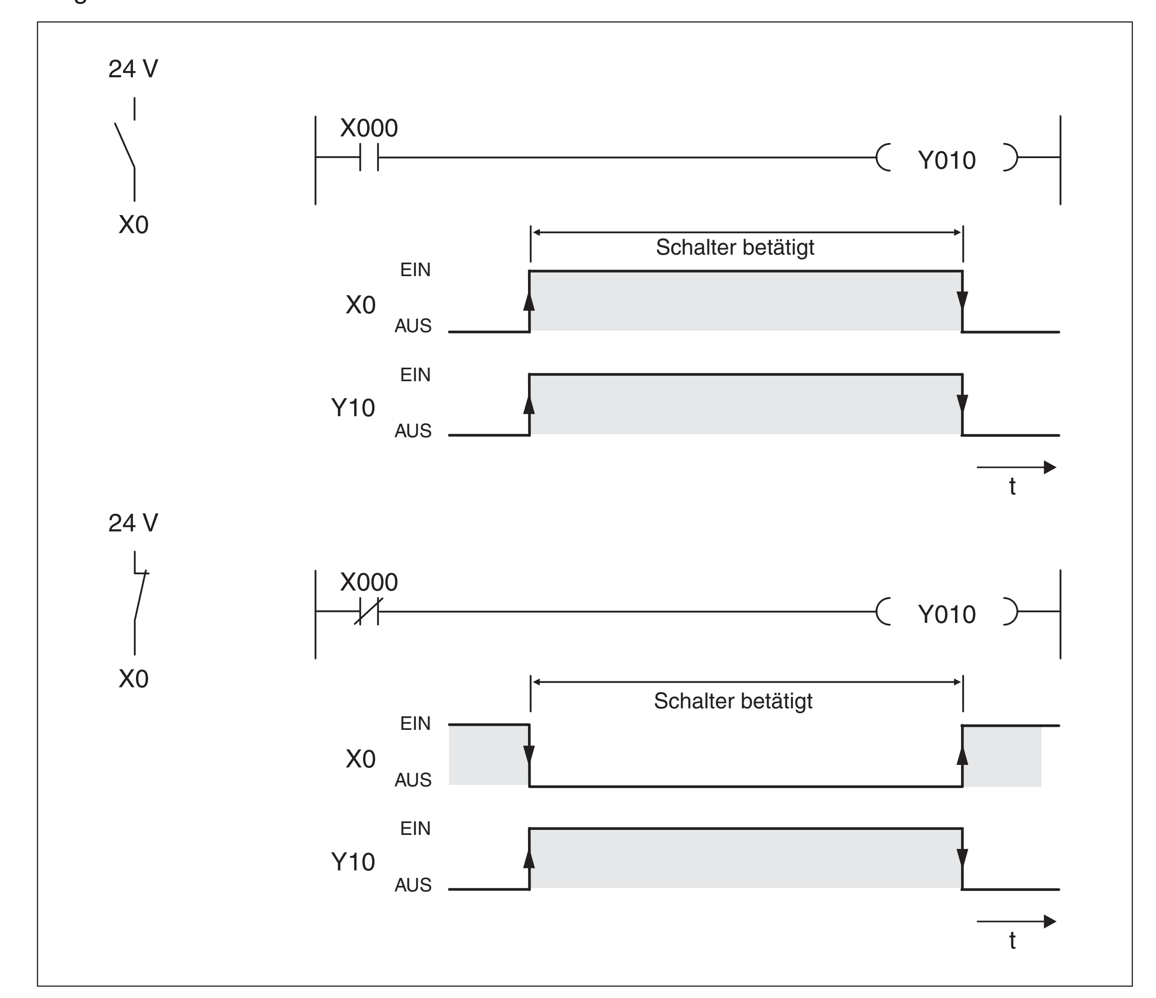
Bei der Programmierung einer SPS muss die Wirkungsweise von Schaltern, Tastern und Sensoren berücksichtigt werden, um die gewünschte Funktion zu erhalten. Eine Steueranweisung prüft – unabhängig von der Art und Weise, wie z. B. ein Eingang angesteuert wird – nur den Signalzustand des angegebenen Eingangs.

| | | |
|---|---|---|
| | Schließer | Bei Betätigung eines Schließers wird der Eingang eingeschaltet (Signalzustand „1“). |
| | Öffner | Bei Betätigung eines Öffners wird der Eingang ausgeschaltet (Signalzustand „0“). |

Schon bei der Programmierung muss daher bekannt sein, ob ein am Eingang der SPS angeschlossener Geber ein Öffner oder ein Schließer ist. Ein Eingang, an dem ein Schließer angeschlossen ist, muss anders behandelt werden, wie ein Eingang mit angeschlossenem Öffner. Das folgende Beispiel soll das verdeutlichen.

Meist werden Geber mit Schließerkontakten verwendet. In einigen Fällen, wie zum Beispiel zum Ausschalten von Antrieben, kommen aber aus Sicherheitsgründen Öffner zum Einsatz (siehe Abschnitt 4.8).

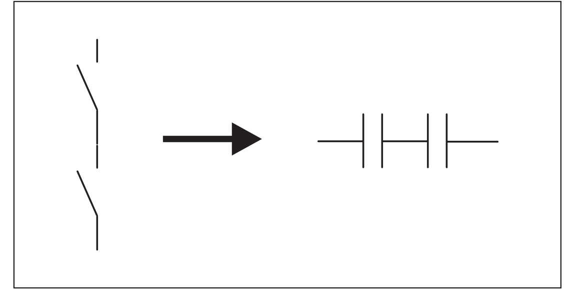
Die folgende Abbildung zeigt zwei Programmsequenzen, bei denen trotz unterschiedlicher Geber das gleiche Ergebnis erreicht wird: Bei Betätigung des Schalters wird der Ausgang eingeschaltet.



## 4.7.4 UND-Verknüpfungen

| Anweisung | Bedeutung | Kontaktplan | IEC-Anweisungsliste |
|---|---|---|---|
| **AND** | UND, (UND-Verknüpfung mit Abfrage auf Signalzustand „1“) | | AND |
| **ANI** | UND-Nicht, (UND-Verknüpfung mit Abfrage auf Signalzustand „0“) | | ANDN |

Eine UND-Verknüpfung entspricht einer Reihenschaltung von mehreren, mindestens aber zwei Schaltern. Nur wenn alle Kontakte geschlossen sind, fließt Strom. Falls ein oder mehrere Kontakte geöffnet sind, ist die UND-Funktion nicht erfüllt, es fließt kein Strom.



In der Programmier-Software werden für die AND- und die ANI-Anweisung übrigens die selben Schaltfelder und Funktionstasten verwendet, wie bei der LD- bzw. der LDI-Anweisung. Bei der Programmierung im Kontaktplan ordnet die Software die Anweisungen entsprechend der Einfügeposition automatisch zu.
Falls Sie in Anweisungsliste programmieren, beachten Sie bitte, dass die AND- und die ANI-Anweisung nicht am Anfang eines Strompfades programmiert werden dürfen. Der Beginn einer Verknüpfung wird mit einer LD- oder LDI-Anweisung programmiert (Abschnitt 4.7.1).

**Beispiel für die AND-Anweisung**

Kontaktplan

AND-Anweisung
X0 X1 Y10

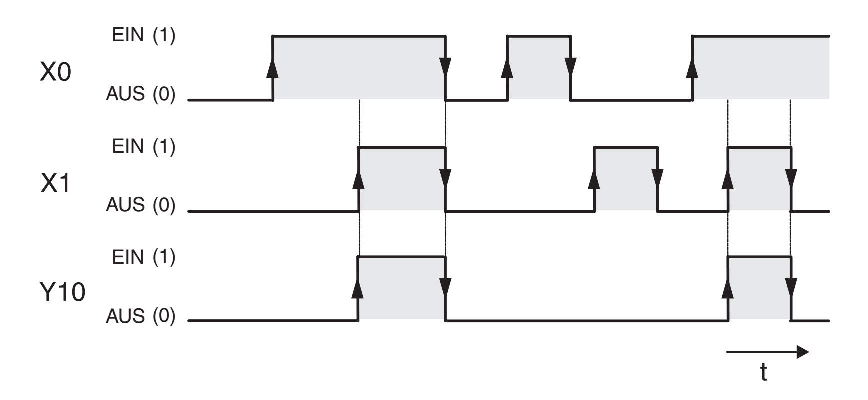
MELSEC-Anweisungsliste

| | |
|---|---|
| LD | X0 |
| AND | X1 |
| OUT | Y10 |

IEC-Anweisungsliste

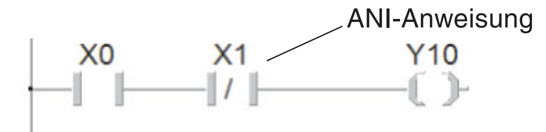
| | |
|---|---|
| LD | X0 |
| AND | X1 |
| ST | Y10 |

Der Ausgang Y10 wird nur eingeschaltet, wenn X0 **und** X1 eingeschaltet sind:



**Beispiel für die ANI-Anweisung**

Kontaktplan



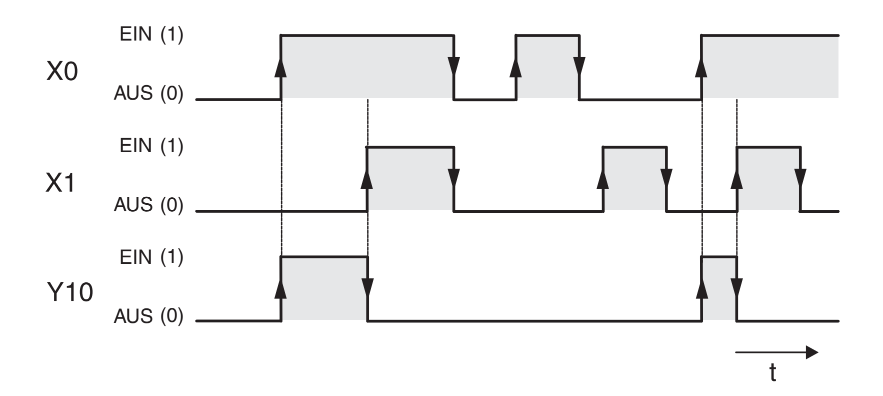

MELSEC-Anweisungsliste

| | |
|---|---|
| LD | X0 |
| ANI | X1 |
| OUT | Y10 |

IEC-Anweisungsliste

| | |
|---|---|
| LD | X0 |
| ANDN | X1 |
| ST | Y10 |

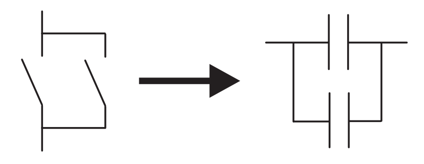
Der Ausgang Y10 wird nur eingeschaltet, wenn X0 eingeschaltet **und** X1 ausgeschaltet ist:




## 4.7.5 ODER-Verknüpfungen

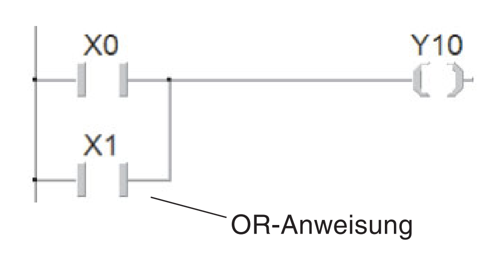
| Anweisung | Bedeutung | Kontaktplan | IEC-Anweisungsliste |
|---|---|---|---|
| **OR** | ODER (ODER-Verknüpfung mit Abfrage auf Signalzustand „1“) | | OR |
| **ORI** | ODER-Nicht, (ODER-Verknüpfung mit Abfrage auf Signalzustand „0“) | | ORN |

Eine ODER-Verknüpfung entspricht in der Schaltungstechnik einer Parallelschaltung von mehreren Schaltern. Sobald ein Kontakt geschlossen ist, fließt Strom. Nur wenn keiner der Kontakte geschlossen ist, fließt kein Strom.



**Beispiel für die OR-Anweisung**

Kontaktplan



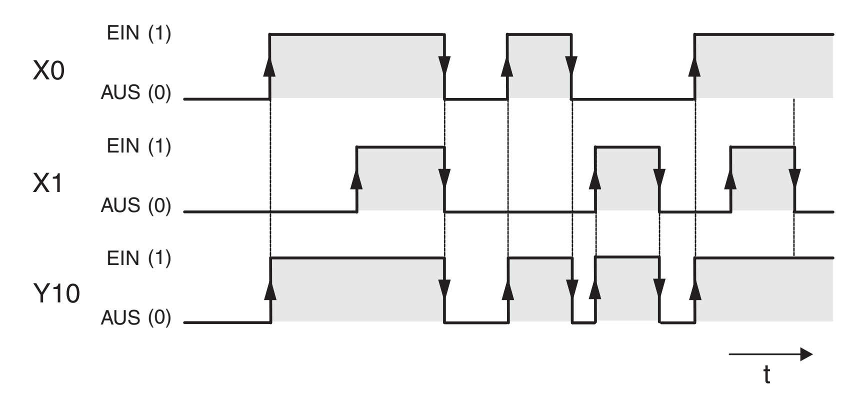

MELSEC-Anweisungsliste

| | |
|---|---|
| LD | X0 |
| OR | X1 |
| OUT | Y10 |

IEC-Anweisungsliste

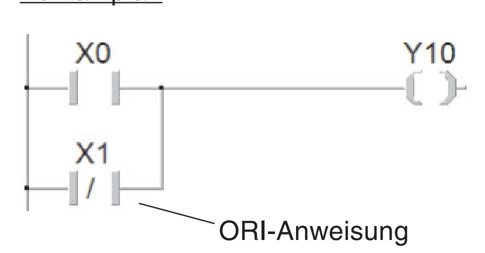
| | |
|---|---|
| LD | X0 |
| OR | X1 |
| ST | Y10 |

Bei diesem Beispiel wird der Ausgang Y10 eingeschaltet, wenn X0 **oder** X1 eingeschaltet ist:




**Beispiel für die ORI-Anweisung**

Kontaktplan



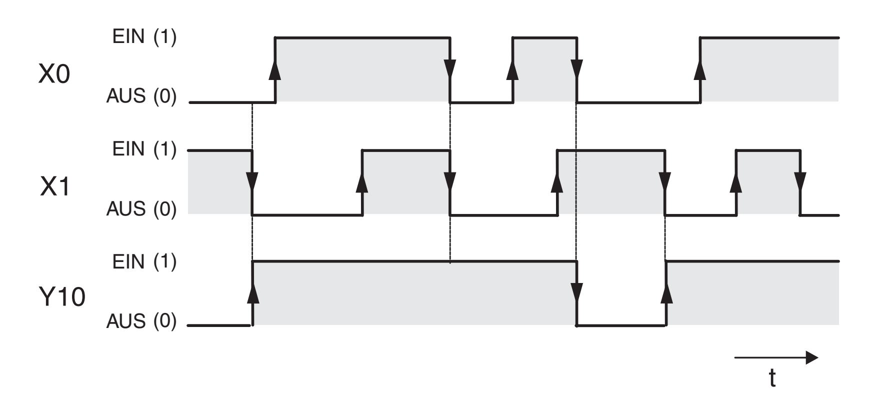

MELSEC-Anweisungsliste

| | |
|---|---|
| LD | X0 |
| ORI | X1 |
| OUT | Y10 |

IEC-Anweisungsliste

| | |
|---|---|
| LD | X0 |
| ORN | X1 |
| ST | Y10 |

Der Ausgang Y10 ist eingeschaltet, wenn X0 eingeschaltet **oder** X1 ausgeschaltet ist:



## 4.7.6 Anweisungen zur Verbindung von Verknüpfungen

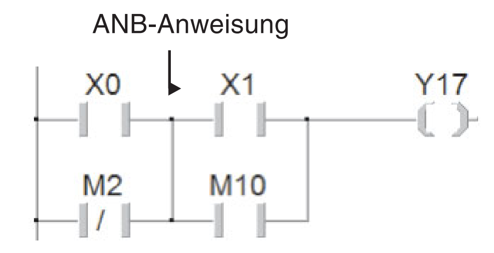
| Anweisung | Bedeutung | Kontaktplan | IEC-Anweisungsliste |
|---|---|---|---|
| **ANB** | UND-Block, (Reihenschaltung von parallelen Verknüpfungen) | ——— | AND ( ... ) |
| **ORB** | ODER-Block (Parallelschaltung von in Reihe geschalteten Verknüpfungen) | \| | OR ( ... ) |

Die ANB- und die ORB-Anweisung sind zwar Anweisungen für die SPS, erscheinen bei der Programmierung im Kontaktplan aber nur als Verbindungslinien. Erst bei der Darstellung oder Programmierung des Programms als Anweisungsliste treten diese Anweisungen in Erscheinung und müssen auch mit ihrer Abkürzung ANB bzw. ORB eingegeben werden.

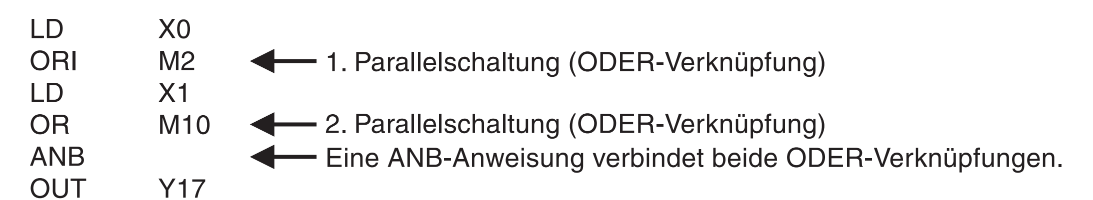
Beide Anweisungen kommen ohne Operanden aus und können beliebig oft im Programm verwendet werden. Die Anzahl der LD- und LDI-Anweisungen und somit auch die Anzahl der ORB- bzw. ANB-Anweisungen vor einer Ausgabeanweisung ist jedoch auf 15 begrenzt.

**Beispiel für die ANB-Anweisung**

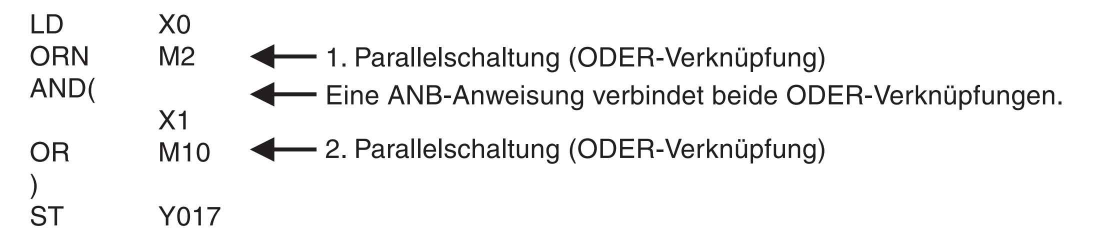
Kontaktplan




MELSEC-Anweisungsliste


LD X0
ORI M2 ← 1. Parallelschaltung (ODER-Verknüpfung)
LD X1
OR M10 ← 2. Parallelschaltung (ODER-Verknüpfung)
ANB ← Eine ANB-Anweisung verbindet beide ODER-Verknüpfungen.
OUT Y17

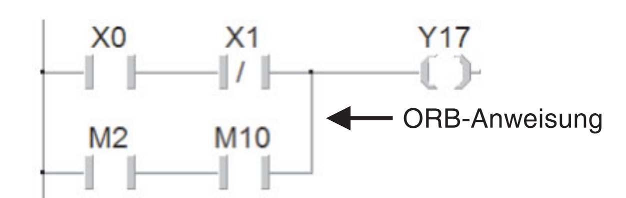
IEC-Anweisungsliste


LD X0
ORN M2 ← 1. Parallelschaltung (ODER-Verknüpfung)
AND( ← Eine ANB-Anweisung verbindet beide ODER-Verknüpfungen.
X1
OR M10 ← 2. Parallelschaltung (ODER-Verknüpfung)
)
ST Y017

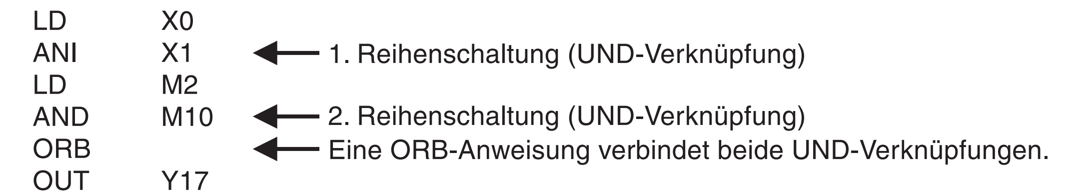
In diesem Beispiel wird der Ausgang Y17 eingeschaltet, wenn der Eingang X0 „1“ **oder** der Merker M2 „0“ ist **und** der Eingang X1 „1“ **oder** der Merker M10 „1“ ist.

**Beispiel für die ORB-Anweisung**

Kontaktplan

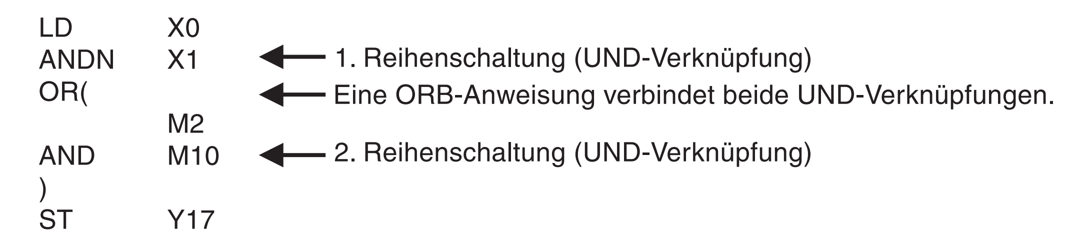



MELSEC-Anweisungsliste



```
LD    X0
ANI   X1   ← 1. Reihenschaltung (UND-Verknüpfung)
LD    M2
AND   M10  ← 2. Reihenschaltung (UND-Verknüpfung)
ORB        ← Eine ORB-Anweisung verbindet beide UND-Verknüpfungen.
OUT   Y17
```

IEC-Anweisungsliste



```
LD    X0
ANDN  X1   ← 1. Reihenschaltung (UND-Verknüpfung)
OR(        ← Eine ORB-Anweisung verbindet beide UND-Verknüpfungen.
      M2
AND   M10  ← 2. Reihenschaltung (UND-Verknüpfung)
)
ST    Y17
```

Der Ausgang Y17 wird eingeschaltet, wenn der Eingang X0 „1“ **und** der Eingang X1 „0“ ist **oder** wenn der Merker M2 „1“ **und** der Merker M10 „1“ ist.

## 4.7.7 Flankengesteuerte Ausführung von Verknüpfungen

| Anweisung | Bedeutung | Kontaktplan | IEC-Anweisungsliste |
|---|---|---|---|
| **LDP** | Lade-Anweisung bei steigender Flanke des Operanden | LDP_M<br>EN ENO<br>s | — |
| **LDF** | Lade-Anweisung bei fallender Flanke des Operanden | LDF_M<br>EN ENO<br>s | — |
| **ANDP** | UND-Verknüpfung bei steigender Flanke des Operanden | ANDP_M<br>EN ENO<br>s | ANDP_M |
| **ANDF** | UND-Verknüpfung bei fallender Flanke des Operanden | ANDF_M<br>EN ENO<br>s | ANDF_M |
| **ORP** | ODER-Verknüpfung bei steigender Flanke des Operanden | ORP_M<br>EN ENO<br>s | ORP_M |
| **ORF** | ODER-Verknüpfung bei fallender Flanke des Operanden | ORF_M<br>EN ENO<br>s | ORF_M |

Im SPS-Programm muss oft die steigende oder fallende Flanke von Operanden erfasst und ausgewertet werden. Bei einer steigenden Flanke wechselt der Signalzustand von „0“ nach „1“ und bei einer fallenden Flanke von „1“ nach „0“.

Verknüpfungen, die auf eine Flanke reagieren, liefern nur in dem Programmzyklus ein „1“-Signal, in dem der abgefragte Operand seinen Signalzustand ändert.

Ohne eine Flankenauswertung liefert zum Beispiel ein Schalter, der an einem Fließband von den vorbeilaufenden Paketen betätigt wird und mit dem die Anzahl der Pakete gezählt werden soll, ein falsches Ergebnis, denn dann erhöht sich der Zählerstand in jedem Programmzyklus um den Wert „1“, solange der Schalter betätigt ist. Wird aber die steigende Flanke des Eingangs erfasst, wird der Zählwert durch jedes Paket nur einmal erhöht.

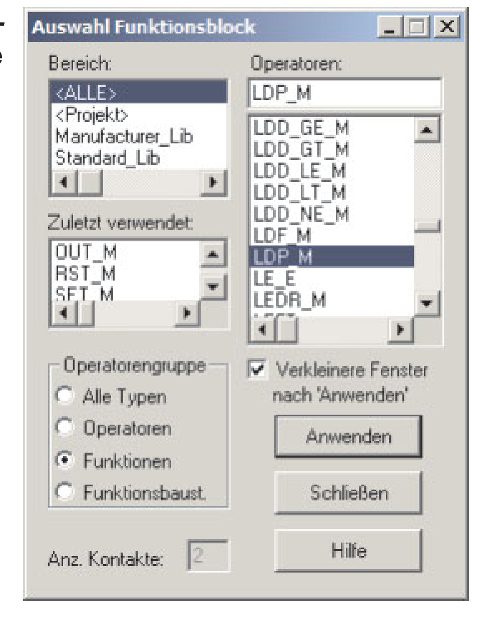
Übrigens können die meisten Applikationsanweisungen ebenfalls flankengesteuert ausgeführt werden (siehe Kap. 6).

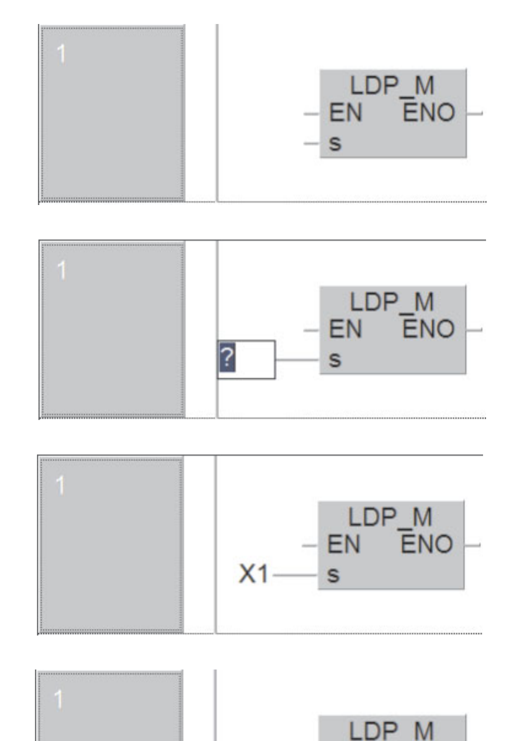
**Intermezzo: Eingabe von Funktionen und Funktionsbausteine in Kontaktplan**

Die flankengesteuerten Anweisungen und andere komplexere Anweisungen können bei der Programmier-Software GX IEC Developer nicht mehr direkt über Schaltfelder in der Werkzeugleiste eingegeben werden. Die Eingabe erfolgt durch Auswahl der Anweisung in einem Dialogfenster.

Klicken Sie zur Eingabe in der Werkzeugleiste auf das Schaltfeld (Funktionsblock). Dadurch wird das unten abgebildete Dialogfenster geöffnet.

Klicken Sie im Feld ***Operatorengruppe*** auf ***Funktionen*** und wählen Sie aus der Liste zum Beispiel die Anweisung LDP_M.



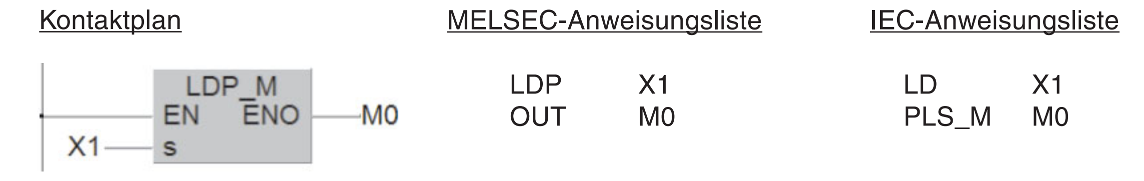

Klicken Sie auf ***Anwenden*** oder doppelt auf das ausgewählte Objekt und dann in den Programmierbereich, um es abzulegen.

Klicken Sie auf das Schaltfeld (Eingangsvariable) in der Werkzeugleiste und anschließend auf den Eingang der Funktion, für den ein Operand angegeben werden soll.

Geben Sie den Eingangsoperanden an und betätigen Sie anschließend die ENTER-Taste.

Zur Angabe eines Operanden am Ausgang der Funktion klicken Sie in der Werkzeugleiste auf das Schaltfeld und dann auf den ENO-Ausgang.

### Auswertung einer steigenden Flanke

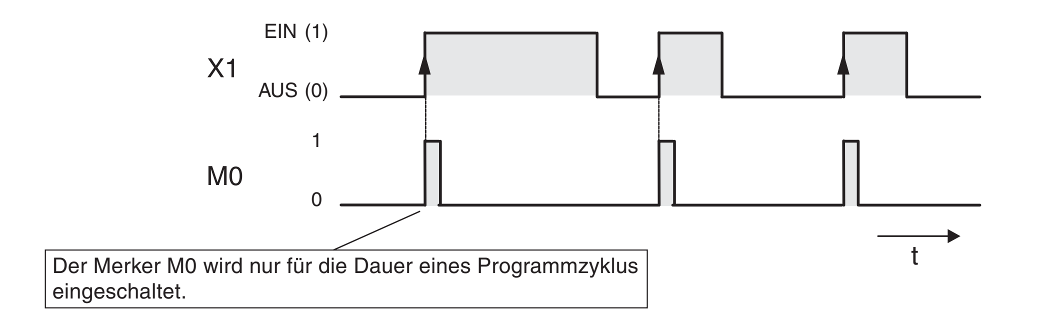



MELSEC-Anweisungsliste

| | |
|---|---|
| LDP | X1 |
| OUT | M0 |

IEC-Anweisungsliste

| | |
|---|---|
| LD | X1 |
| PLS_M | M0 |




### Auswertung einer fallenden Flanke

Kontaktplan

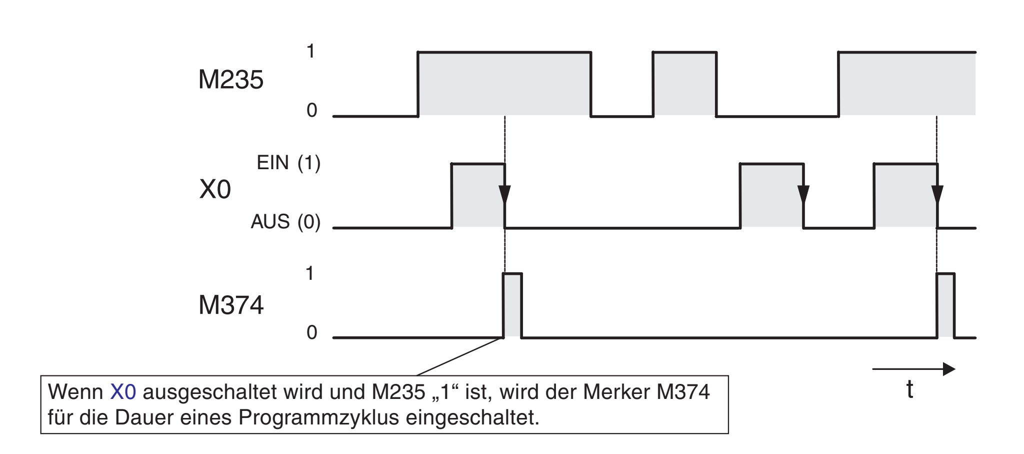
M235 — ANDF_M — EN — ENO — M374; X0 — s

MELSEC-Anweisungsliste

| | |
|---|---|
| LD | M235 |
| ANDF | X0 |
| OUT | M374 |

IEC-Anweisungsliste

| | |
|---|---|
| LD | M235 |
| ANDF_M | X0 |
| ST | M374 |




Bis auf die Flankenauswertung ist die Funktion der LDP- und LDF-Anweisung, der ANDP- und ANDF-Anweisung sowie der ORP- und ORF-Anweisung identisch mit der LD-, der AND- bzw. der OR-Anweisung, d.h. die flankengesteuerten Anweisungen können genau so im Programm verwendet werden, wie die „normalen“ Anweisungen.

## 4.7.8 Setzen und Rücksetzen

| Anweisung | Bedeutung | Kontaktplan | IEC-Anweisungsliste |
|---|---|---|---|
| SET | Setzen eines Operanden①, (Zuweisung von Signalzustand „1“) | -(S)- SET_M EN ENO d | S |
| RST | Rücksetzen eines Operanden②, (Zuweisung von Signalzustand „0“) | -(R)- RST_M EN ENO d | R |

① Mit einer SET-Anweisung können Ausgänge (Y), Merker (M) und Schrittmerker (S) gesetzt werden.

② Mit einer RST-Anweisung können Ausgänge (Y), Merker (M),Schrittmerker (S), Timer (T), Counter (C) und Register (D, V, Z) zurückgesetzt werden.

Der Signalzustand einer OUT-Anweisung ist nur solange „1“, wie auch das Verknüpfungsergebnis vor der OUT-Anweisung „1“ ist. Wird zum Beispiel an einem Eingang ein Taster angeschlossen und an einem Ausgang eine Lampe, brennt bei der Kombination einer LD- und einer OUT-Anweisung die Lampe nur, solange der Taster betätigt wird.

Mit einer SET-Anweisung wird ein Ausgang oder Merker nach einem kurzem Einschaltimpuls eingeschaltet (= gesetzt). Der Operand bleibt dann solange eingeschaltet, bis er durch eine RST-Anweisung wieder ausgeschaltet (= zurückgesetzt) wird. So lassen sich z. B. Selbsthaltungen oder das Ein- und Ausschalten von Antrieben mit Hilfe von Drucktastern realisieren. (Ein Ausgang wird auch ausgeschaltet, wenn die SPS gestoppt oder die Versorgungsspannung ausgeschaltet wird. Einige der Merkern behalten auch in diesen Fällen ihren letzten Signalzustand, bleiben also zum Beispiel gesetzt.)

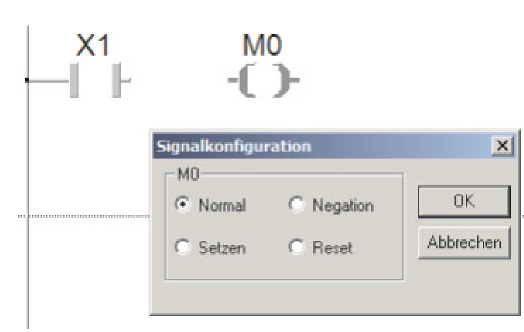
Bei der Programmierung im Kontaktplan können SET- und RST-Anweisungen innerhalb von OUT-Anweisungen oder als Funktionen programmiert werden.

### OUT-Anweisung mit Setz- oder Rücksetzfunktion

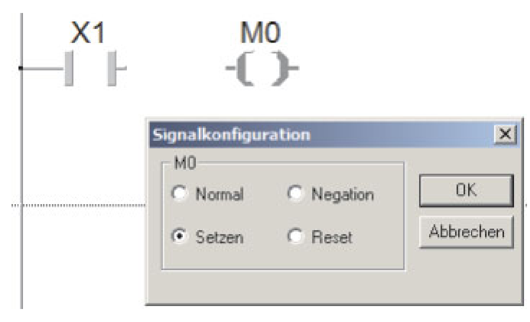
Programmieren Sie eine OUT-Anweisung und geben Sie den Operanden an, der gesetzt oder zurückgesetzt werden soll.

X1 M0

Klicken Sie anschließend doppelt auf die OUT-Anweisung. Es öffnet sich das Dialogfenster ***Signalkonfiguration***.



Für eine SET-Anweisung klicken Sie im Dialogfeld auf ***Setzen***. Falls eine RST-Anweisung programmiert werden soll, klicken Sie bitte auf ***Reset***. Schließen Sie das Fenster durch einen Mausklick auf das Schaltfeld ***OK***.



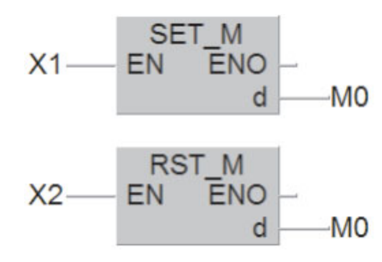

Die Konvertierung der OUT-Anweisung in eine SET-Anweisung ist damit abgeschlossen.




**Beispiele zur Setz-/Rücksetzfunktion**

Kontaktplan

1. Variante




2. Variante

X1 — SET_M EN ENO d — M0

X2 — RST_M EN ENO d — M0

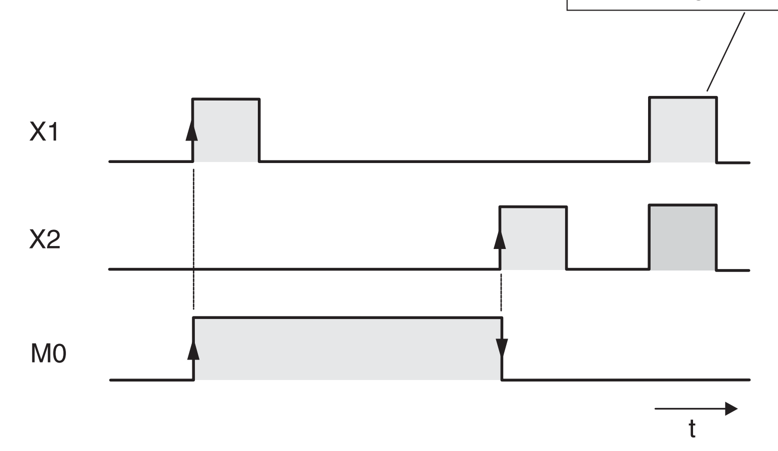
MELSEC-Anweisungsliste

| | |
|---|---|
| LD | X1 |
| SET | M0 |
| LD | X2 |
| RST | M0 |

IEC-Anweisungsliste

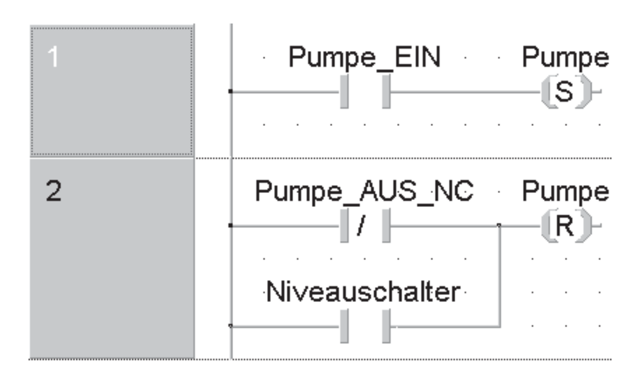
| | |
|---|---|
| LD | X1 |
| S | M0 |
| LD | X2 |
| R | M0 |

Wenn die Setz- und die Rücksetzanweisung eines Operanden im selben Zyklus „1“ sind, hat die in der Reihenfolge letzte Operation Vorrang. In diesem Beispiel ist das die RST-Anweisung, M0 wird nicht gesetzt.



Als Beispiel für eine Anwendung ist eine Pumpensteuerung zur Füllung eines Behälters dargestellt. Die Pumpe kann mit den Tastern „EIN“ und „AUS“ manuell gesteuert werden. Aus Sicherheitsgründen wird zum Ausschalten ein Taster mit Öffnerkontakt verwendet. Wenn der Behälter gefüllt ist, schaltet ein Niveauschalter die Pumpe aus.

Kontaktplan




MELSEC-Anweisungsliste

| | |
|---|---|
| LD | Pumpe_EIN |
| SET | Pumpe |
| LDI | Pumpe_AUS_NC |
| OR | Niveauschalter |
| RST | Pumpe |

IEC-Anweisungsliste

| | |
|---|---|
| LD | Pumpe_EIN |
| S | Pumpe |
| LDN | Pumpe_AUS_NC |
| OR | Niveauschalter |
| R | Pumpe |

**HINWEIS**

Damit die Operanden im Programm mit ihren Bezeichnern dargestellt werden, ist eine Festlegung in der Globalen Variablenliste notwendig. Die folgende Abbildung zeigt die globale Variablenliste für dieses Beispiel:

| | Klasse | Bezeichner | MIT-Adr. | IEC-Adress | Typ |
|---|---|---|---|---|---|
| 0 | VAR_GLOBAL | Pumpe_EIN | X1 | %IX1 | BOOL |
| 1 | VAR_GLOBAL | Pumpe_AUS_NC | X2 | %IX2 | BOOL |
| 2 | VAR_GLOBAL | Niveauschalter | X3 | %IX3 | BOOL |
| 3 | VAR_GLOBAL | Pumpe | Y10 | %QX16 | BOOL |

Weitere Informationen zur Globalen Variablenliste finden Sie in Abschnitt 4.6.2.

## 4.7.9 Erzeugung eines Impulses

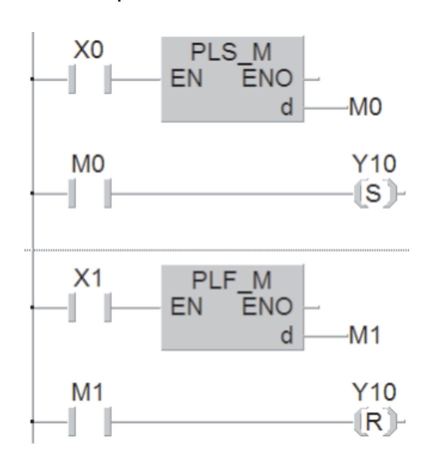
| Anweisung | Bedeutung | Kontaktplan | IEC-Anweisungsliste |
|---|---|---|---|
| **PLS** | Setzen eines Operanden* für die Dauer eines Programmzyklus bei steigender Flanke der Eingangsbedingung | PLS_M EN ENO d | PLS_M |
| **PLF** | Setzen eines Operanden* für die Dauer eines Programmzyklus bei fallender Flanke der Eingangsbedingung | PLF_M EN ENO d | PLF_M |

* Mit einer PLS- oder PLF-Anweisung können Ausgänge (Y) und Merker (M) gesteuert werden.

Wird eine PLS-Anweisung an Stelle einer OUT-Anweisung verwendet, hat der angegebene Operand nur in dem Programmzyklus den Signalzustand „1“, in dem der Signalzustand der Verknüpfungen vor der PLS-Anweisung von „0“ nach „1“ wechselt (steigende Flanke).

Eine PLF-Anweisung reagiert bei der fallenden Flanke und liefert für einen Programmzyklus den Signalzustand „1“, wenn der Signalzustand der Verknüpfungen vor dieser Anweisung von „1“ nach „0“ wechselt

Kontaktplan

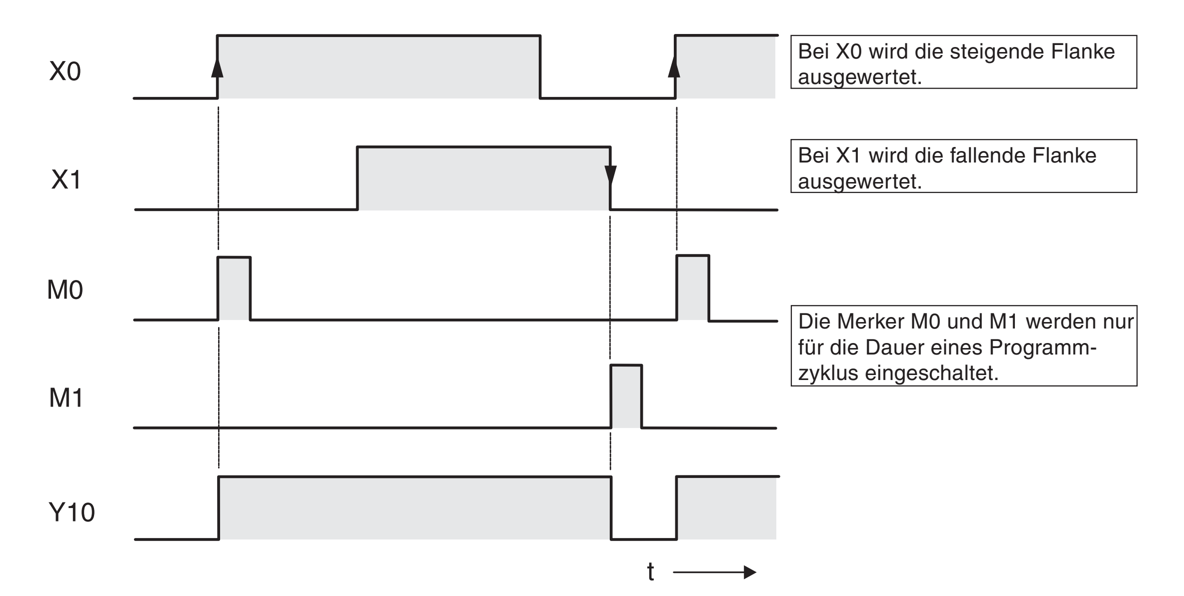



MELSEC-Anweisungsliste

```
LD    X0
PLS   M0
LD    M0
SET   Y10
LD    X1
PLF   M1
LD    M1
RST   Y10
```

IEC-Anweisungsliste

```
LD     X0
PLS_M  M0
LD     M0
S      Y10
LD     X1
PLF_M  M1
LD     M1
R      Y10
```



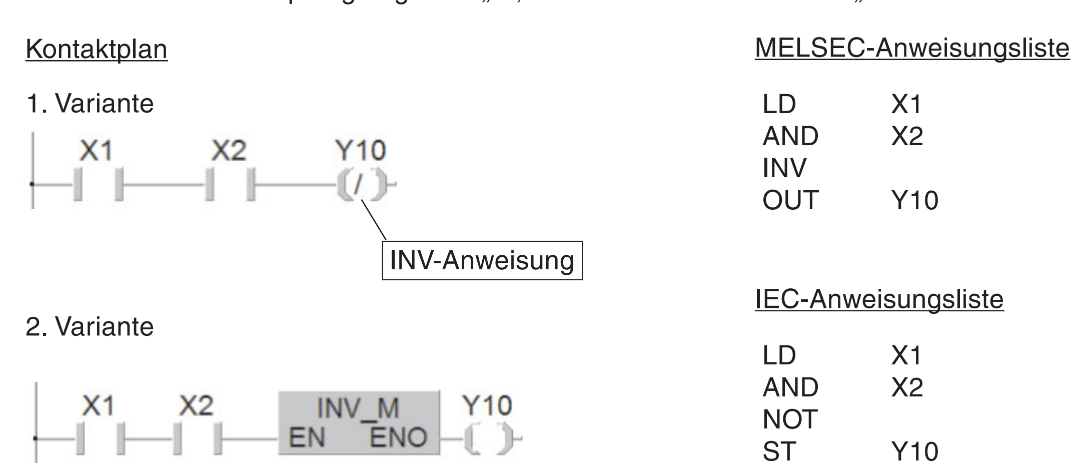
## 4.7.10 Verknüpfungsergebnis umkehren

| Anweisung | Bedeutung | Kontaktplan | IEC-Anweisungsliste |
|---|---|---|---|
| INV | Umkehrung eines Verknüpfungsergebnisses | -( / )- <br> INV_M EN ENO | NOT |

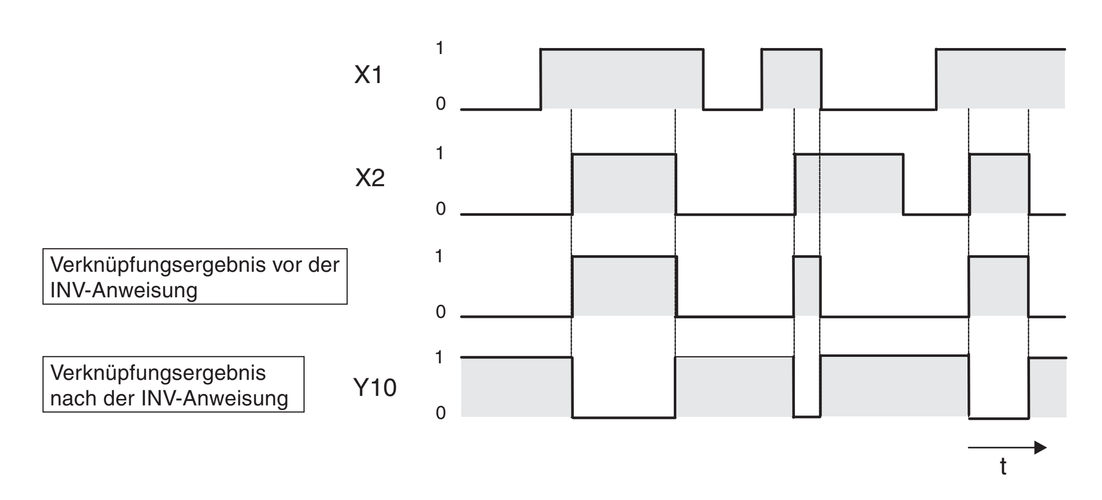
Eine INV-Anweisung wird ohne Operanden angegeben und kehrt das Verknüpfungsergebnis, das vor der Ausführung der INV-Anweisung gültig war, um:

– Lautet das Verknüpfungsergebnis „1“, wird es nach der Inversion „0“.

– Lautet das Verknüpfungsergebnis „0“, wird es nach der Inversion „1“.



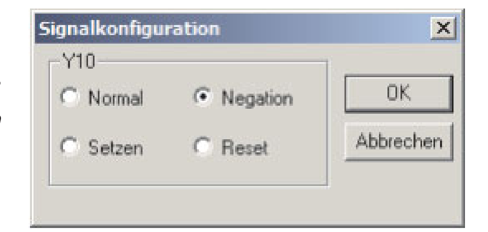

Für das oben gezeigte Beispiel ergibt sich der folgende Signalverlauf:




Die INV-Anweisung kann verwendet werden, wenn das Ergebnis einer komplexen Verknüpfung umgekehrt werden muss.

**HINWEIS**

Um im Kontaktplan eine INV-Anweisung innerhalb einer OUT-Anweisung zu programmieren, klicken Sie doppelt auf die OUT-Anweisung. Im Dialogfenster ***Signalkonfiguration*** wählen Sie dann ***Negation*** (siehe auch Abschnitt 4.7.8)



## 4.7.11 Zustand eines Bit-Ausgangsoperanden umkehren

| Anweisung | Bedeutung | Kontaktplan | IEC-Anweisungsliste |
|---|---|---|---|
| **FF** | Invertierung eines Bit-Ausgangs-operanden* | FF_MD EN ENO s | FF_MD |

* Mit einer FF-Anweisung können Ausgänge (Y), Merker (M) und auch einzelne Bits von Wordoperanden gesteuert werden.

Die FF-Anweisung kehrt den Signalzustand des mit der Anweisung angegebenen Operanden bei steigender Flanke am Eingang der FF-Anweisung um.

– War der Zustand des Operanden „1“, wird er nach der Ausführung der FF-Anweisung „0“.

– War der Zustand des Operanden „0“, wird er nach der Ausführung der FF-Anweisung „1“.

Kontaktplan

X1
FF_MD
EN ENO
s
Y10

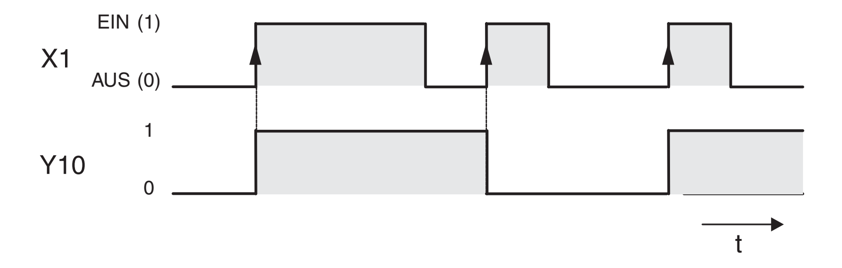
MELSEC-Anweisungsliste

| | |
|---|---|
| LD | X1 |
| FF | Y10 |

IEC-Anweisungsliste

| | |
|---|---|
| LD | X1 |
| FF_MD | Y10 |

Im Beispiel oben wird bei jedem Einschalten des Eingangs X1 der Zustand des Ausgangs Y10 verändert:

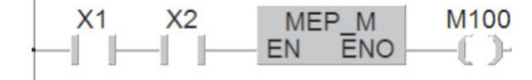


## 4.7.12 Umwandlung von Verknüpfungsergebnissen in Impulse

| Anweisung | Bedeutung | Kontaktplan | IEC-Anweisungsliste |
|---|---|---|---|
| **MEP** | Pulserzeugung bei ansteigender Flanke des Verknüpfungsergebnisses | MEP_M EN ENO | MEP_M |
| **MEF** | Pulserzeugung bei abfallender Flanke des Verknüpfungsergebnisses | MEF_M EN ENO | MEF_M |

Die Anweisungen MEP und MEF werden ohne Operanden angegeben. Sie erzeugen aus der steigenden bzw. fallenden Flanke des Verknüfungsergebnisses, das vor der Ausführung dieser Anweisungen gültig war, einmalig einen Impuls. Der nächste Impuls wird erst bei einer erneuten Flanke erzeugt.

Kontaktplan



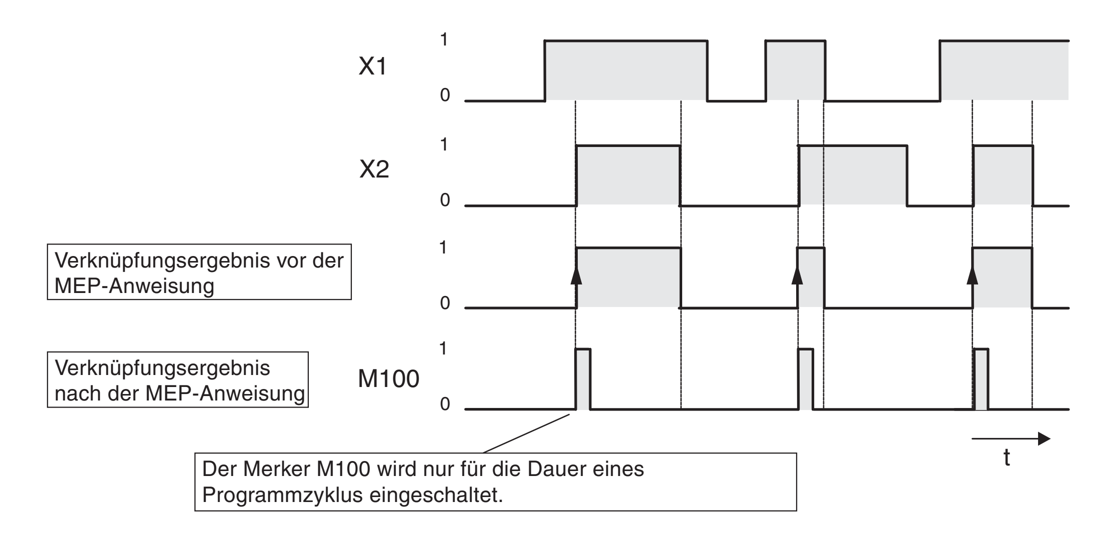

MELSEC-Anweisungsliste

```
LD      X1
AND     X2
MEP
OUT     M100
```

IEC-Anweisungsliste

```
LD      X1
AND     X2
MEP_M
ST      M100
```

Den Signalverlauf für dieses Beispiel zeigt die folgende Abbildung :




Die MEP- und MEF-Anweisungen eignen sich besonders bei der Verwendung von mehreren zusammengeschalteten Kontakten. Mehrere in Reihe geschaltete Schließer haben beispielsweise im betätigten Zustand ständig eine 1 als Verknüpfungsergebnis. Wird dadurch ein Merker gesetzt, könnte dieser nicht an anderer Stelle im Programm zurückgesetzt werden. Durch die Reihenschaltung mit einer MEP-Anweisung ist das Rücksetzen möglich, da nur dann ein Impuls erzeugt wird, wenn das Verknüpfungsergebnis der Reihenschaltung von 0 auf 1 wechselt.

## 4.8 Sicherheit geht vor!

Eine SPS hat zwar viele Vorteile gegenüber einer festverdrahtenden Steuerung, bei Fragen der Sicherheit darf man sich aber nicht vollständig auf sie verlassen.

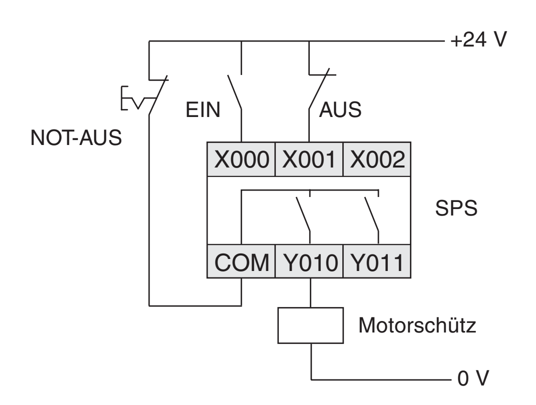
### NOT-AUS-Einrichtungen

Durch Fehler in der Steuerung einer Anlage dürfen weder Personen noch Maschinen gefährdet werden. Deshalb müssen NOT-AUS-Einrichtungen auch wirksam sein, wenn die SPS nicht mehr korrekt arbeitet und z. B. die Spannungsversorgung der SPS-Ausgänge abschalten.

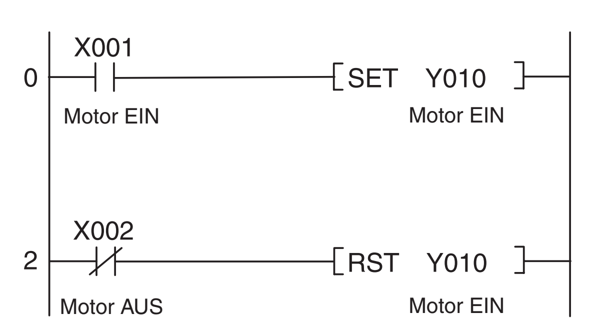
Auf keinen Fall darf ein NOT-AUS-Taster nur als Eingang in der SPS verarbeitet und die Abschaltung durch das Programm ausgelöst werden.

### Sicherheit auch bei Drahtbruch

Die Betriebssicherheit muss auch gewährleistet sein, wenn die Übertragung der Signale von den Schaltern zur SPS unterbrochen wird. Aus diesem Grund werden Einschaltbefehle durch Schalter oder Taster mit Schließerkontakten und Ausschaltbefehle mit Öffnerkontakten zur SPS übermittelt.




In diesem Beispiel kann das Schütz für einen Antrieb zusätzlich durch einen NOT-AUS-Schalter ausgeschaltet werden.



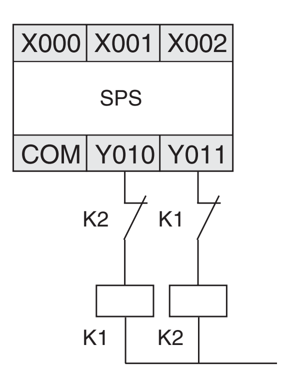

Im Programm wird der Schließerkontakt des EIN-Tasters mit einer LD-Anweisung und der Öffnerkontakt des AUS-Tasters mit einer LDI-Anweisung abgefragt. Der Ausgang und damit der Antrieb wird abgeschaltet, wenn der Eingang X002 den Signalzustand „0“ hat. Das ist der Fall bei der Betätigung des AUS-Tasters oder wenn die Verbindung zwischen dem Taster und dem Eingang X002 unterbrochen wird.

Dadurch wird auch bei einem Drahtbruch der Antrieb abgeschaltet bzw. das Einschalten verhindert. Außerdem hat das Ausschalten Vorrang, weil es im Programm nach dem Einschalten bearbeitet wird.

### Verriegelungskontakte

Falls bei einer Schaltung zwei Ausgänge nicht gleichzeitig eingeschaltet werden dürfen, wie z. B. bei der Umschaltung der Drehrichtung von Antrieben, muss diese Verriegelung auch durch Kontakte der angesteuerten Schütze erfolgen. Im Programm findet nur eine interne Verriegelung statt und bei einem Fehler der SPS könnten beide Ausgänge gleichzeitig eingeschaltet werden.

Beispiel für eine Verriegelung durch Schützkontakte: Die Schütze K1 und K2 können nicht zusammen eingeschaltet werden.



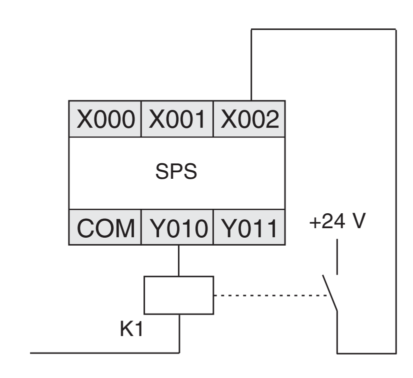

**Zwangsabschaltungen**

Wenn durch eine SPS Bewegungsabläufe gesteuert werden und durch das Überfahren eines Endpunktes Gefahren auftreten können, müssen zusätzliche Endschalter vorgesehen werden, die in diesem Fall die Bewegung unmittelbar und unabhängig von der SPS unterbrechen. Ein Beispiel für eine Zwangsabschaltung finden Sie im Abschnitt 4.9.1.

**Signalrückführungen**

In der Regel werden Ausgänge der SPS nicht überwacht. Ein Ausgang wird eingeschaltet und im Programm wird davon ausgegangen, das außerhalb der SPS die gewünschte Reaktion eintritt. In den meisten Fällen ist das ausreichend. Bei sensiblen Anwendungen jedoch, bei denen Fehler im Ausgangskreis, wie Drahtbrüche oder verschweißte Schütze, schwerwiegende Folgen für die Sicherheit oder die Funktion haben können, sollten auch die ausgegebenen Signale der SPS überwacht werden.

In diesem Beispiel schaltet ein Schließerkontakt von Schütz K1 den Eingang X002 ein, wenn der Ausgang Y010 eingeschaltet wird. So kann im Programm überwacht werden, ob dieser Ausgang und das angeschlossene Schütz korrekt arbeiten.
Nicht erfasst wird, ob die geschaltete Last sich wie gewünscht verhält (z. B. ob sich ein Antrieb tatsächlich dreht). Dazu sind weitere und teilweise aufwendige Kontrollen notwendig, wie beispielsweise eine Überwachung der Lastspannung oder Drehwächter.



# 4.9 Umsetzung einer Steuerungsaufgabe

Eine speicherprogrammierbare Steuerung bietet Ihnen fast unbegrenzte Möglichkeiten zur Verknüpfung von Ein- und Ausgängen. Bei der Vielzahl von Anweisungen, die die Steuerungen des MELSEC System Q bieten, gilt es die zur Lösung einer Steuerungsaufgabe geeigneten Anweisungen auszuwählen und mit ihnen das Programm zu realisieren.

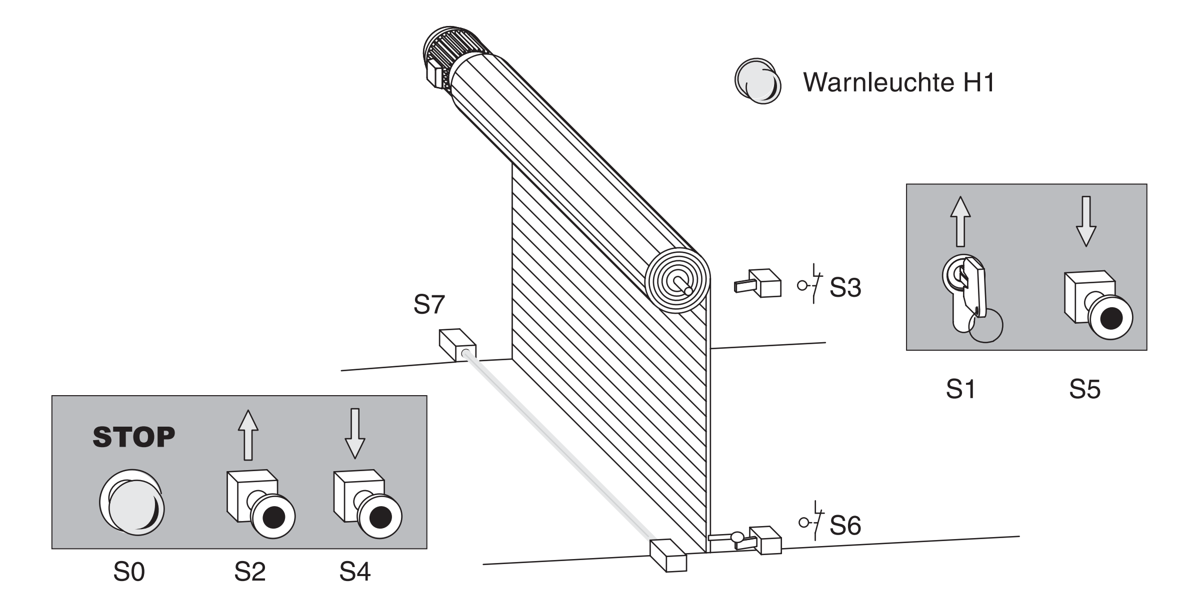
Anhand einer einfachen Steuerungsaufgabe soll der Weg von der Aufgabenstellung bis zum fertigen Programm aufgezeigt werden.

## 4.9.1 Steuerung eines Rolltors

Schon vor der Programmierung muss die Aufgabenstellung klar sein. Man fängt sozusagen „von hinten“ an und beschreibt, was die SPS leisten soll:

**Funktionsbeschreibung**

Ein Rolltor für den Zugang zu einer Lagerhalle soll so gesteuert werden, das die Bedienung von außen oder von innen komfortabel möglich ist. Dabei sollen aber auch Sicherheitsaspekte berücksichtigt werden.




- Bedienung
  - Von außen soll das Tor mit dem Schlüsseltaster S1 geöffnet und mit dem Taster S5 geschlossen werden können. In der Halle soll nach einer Betätigung des Tasters S2 das Tor öffnen und des Tasters S4 das Tor schließen.
  - Eine zusätzliche Zeitsteuerung soll das Tor auch automatisch schließen, wenn es länger als 20 s geöffnet war.
  - Die Zustände „Tor in Bewegung” und „Tor steht in undefinierter Position” sollen von einer blinkende Warnleuchte angezeigt werden.
- Sicherheitseinrichtungen

– Durch einen Stopp-Schalter (S0) soll eine Bewegung des Tores jederzeit angehalten werden können und das Tor in seiner aktuellen Position bleiben. Diese Stopp-Schalter hat keine NOT-AUS-Funktion! Aus diesem Grund wird der Schalter nur in der SPS verarbeitet und schaltet keine externen Spannungen.

– Wenn eine Lichtschranke (S7) beim Schließen des Tores ein Hindernis erkennt, soll das Tor automatisch geöffnet werden.

– Zum Stoppen des Motors in den beiden Endpositionen des Tores sind die beiden Endschalter S3 („Tor ist offen") und S6 („Tor ist geschlossen") vorgesehen.

### Zuordnung der Ein- und Ausgangssignale

Durch die Funktionsbeschreibung ergibt sich schon die Zahl der benötigten Ein- und Ausgänge. Zur Ansteuerung des Antriebsmotors erfolgt über zwei Ausgänge. Die Signale werden den Ein- und Ausgängen der SPS zugeordnet:

| Funktion | | Kenn-zeichen | Adresse | Bemerkung |
|---|---|---|---|---|
| Eingänge | Tastschalter STOP | S0 | X0 | Öffnerkontakt (Bei Betätigung des Schalters ist X0 = „0" und das Tor stoppt.) |
| | Schlüsselschalter Tor AUF (außen) | S1 | X1 | Schließerkontakte |
| | Taster Tor AUF (innen) | S2 | X2 | |
| | Endschalter oben (TOR geöffnet) | S3 | X3 | Öffnerkontakt (X2 = „0", wenn das Tor oben und S3 betätigt ist.) |
| | Taster Tor ZU (innen) | S4 | X4 | Schließerkontakte |
| | Taster Tor ZU (außen) | S5 | X5 | |
| | Endschalter unten (Tor ZU) | S6 | X6 | Öffnerkontakt (X6 = „0", wenn das Tor unten und S6 betätigt ist.) |
| | Lichtschranke | S7 | X7 | X7 wird „1", wenn ein Hindernis erkannt wird |
| Ausgänge | Warnleuchte | H1 | Y10 | — |
| | Motorschütz (Motor Linkslauf) | K1 | Y11 | Linkslauf = Tor öffnen |
| | Motorschütz (Motor Rechtslauf) | K2 | Y12 | Rechtslauf = Tor schließen |
| Zeitgeber | Verzögerung für automatisches Schließen | — | T0 | Zeit: 20 Sekunden |

## 4.9.2 Programmierung

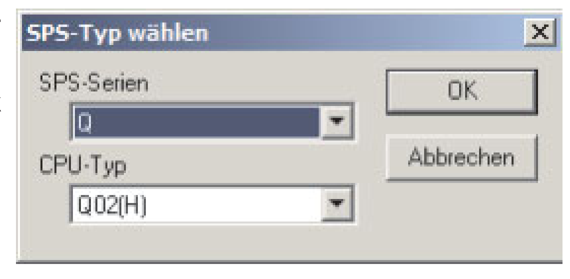
### Neues Projekt anlegen

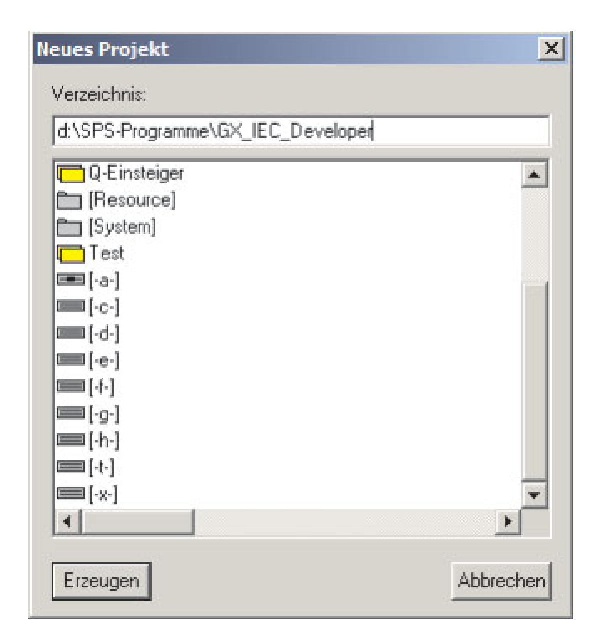
Nach dem Start des GX IEC Developer wählen Sie im Menü ***Projekt*** den Eintrag ***Neu***.




Wählen Sie als SPS-Typ das MELSEC System Q und die verwendete CPU aus.

Bestätigen Sie die Eingabe durch einen Klick auf das Feld ***OK***.

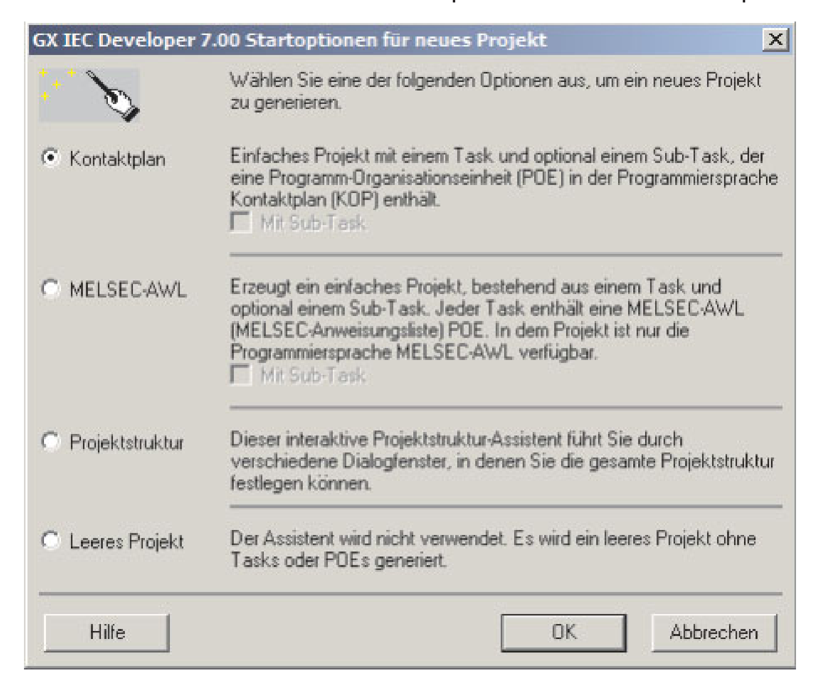




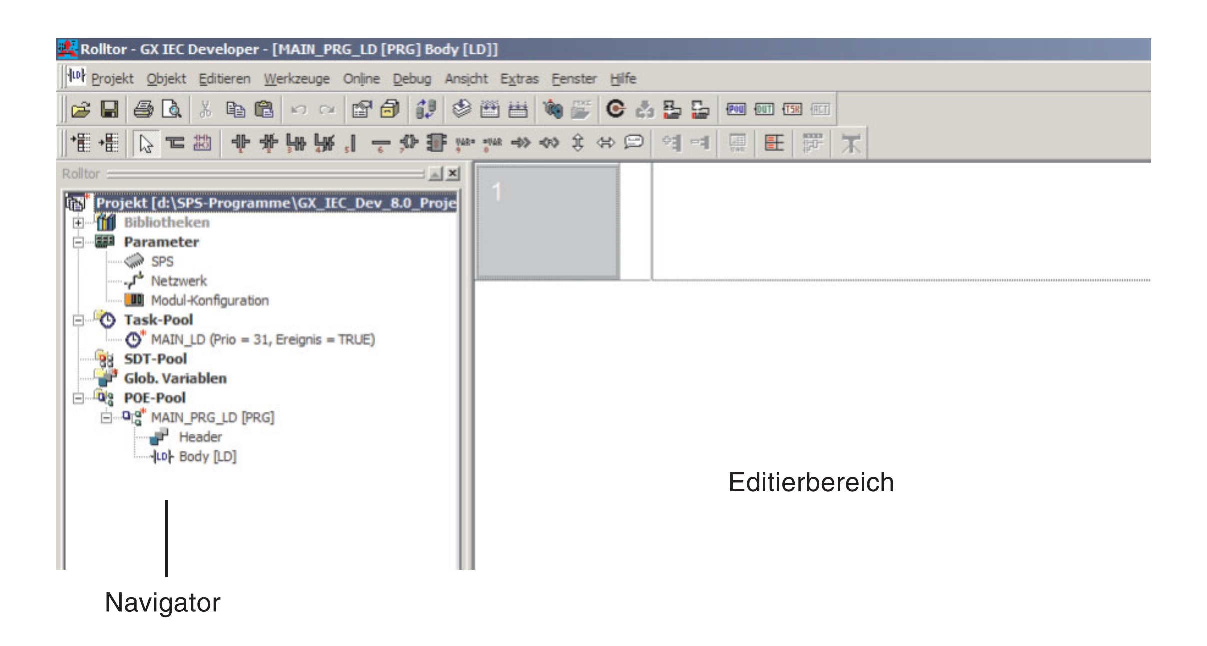

Das Dialogfenster ***Neues Projekt*** öffnet sich nun automatisch. Geben Sie hinter dem Pfad den Namen des neuen Projekts ein.

Nach Klicken auf das Feld ***Erzeugen*** legt GX IEC Developer ein Unterverzeichnis mit dem angegebenen Namen an

Wählen Sie anschließend die Startoptionen. Für dieses Beispiel wird ***Kontaktplan*** gewählt.

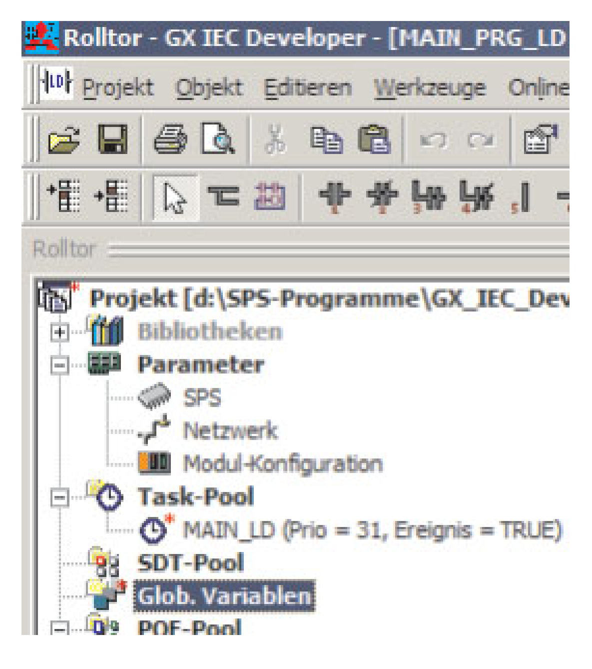



Nach der Bestätigung mit ***OK*** kann programmiert werden. Angezeigt wird der noch leere Body der POE MAIN (siehe Abbildung auf der nächsten Seite).



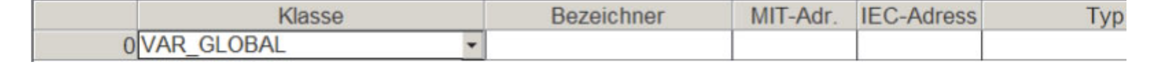

### Definieren der Globalen Variablen

Die Globale Variablenliste muss nicht ausgefüllt werden, wenn im Programm keine symbolischen Namen, sondern nur Mitsubishi-Adressen verwendet werden. Allerdings entspricht das Programm dann nicht mehr der IEC 6113-3.



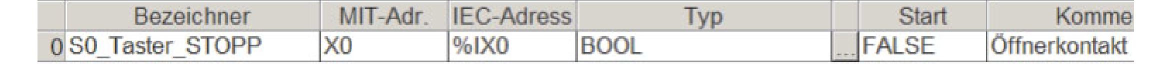

Klicken Sie im Navigator doppelt auf den Zweig ***Glob. Variablen***.

Das Fenster mit der Deklarationstabelle zum Anlegen der Globalen Variablen wird geöffnet.



| | Klasse | Bezeichner | MIT-Adr. | IEC-Adress | Typ |
|---|---|---|---|---|---|
| 0 | VAR_GLOBAL | | | | |

Geben Sie den Bezeichner und die absolute Adresse der ersten Globalen Variablen ein.Die Angabe der absoluten Adresse ist nur in einem Feld erforderlich (MIT-Adr. oder IEC-Adresse). Das jeweils andere Feld wird vom GX IEC Developer automatisch ausgefüllt.

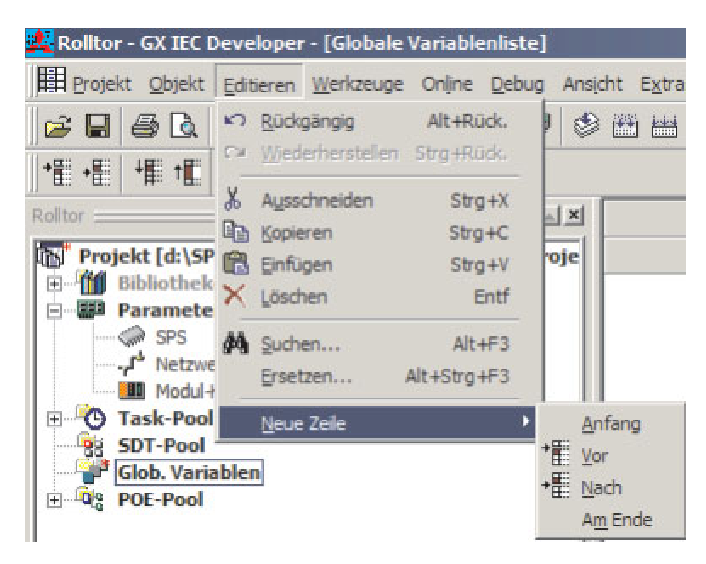


| | Bezeichner | MIT-Adr. | IEC-Adress | Typ | | Start | Komme |
|---|---|---|---|---|---|---|---|
| 0 | S0_Taster_STOPP | X0 | %IX0 | BOOL | ... | FALSE | Öffnerkontakt |

Bei der Eingabe der Adresse eines Eingangs wird als Typ automatisch BOOL eingetragen.

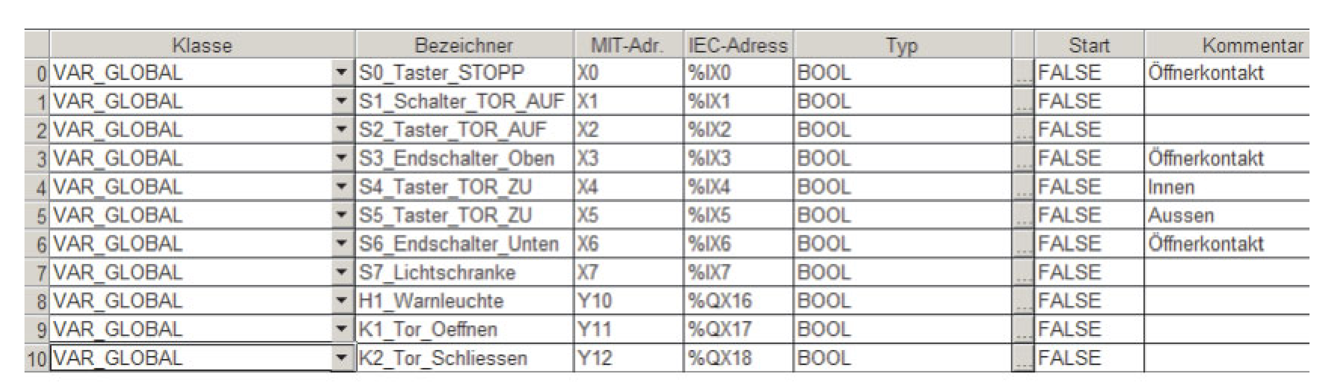
Zur Eingabe weiterer Globaler Variablen muss die Liste erweitert werden. Dazu gibt es mehrere Möglichkeiten:

- Wenn sich der Cursor in einer beliebigenSpalte der letzten Zeile befindet, betätigen Sie die SHIFT-Taste und die ENTER-Taste gleichzeitig.
- Oder wählen Sie im Menü ***Editieren*** eine ***Neue Zeile***.




- Oder klicken Sie in der Werkzeugleiste auf das Schaltfeld „Davor einfügen“ oder „Dahinter einfügen“ .

Nach der Eingabe aller verwendten Ein- und Ausgänge sollte die Liste der Globalen Variablen so aussehen:



| | Klasse | Bezeichner | MIT-Adr. | IEC-Adress | Typ | | Start | Kommentar |
|---|---|---|---|---|---|---|---|---|
| 0 | VAR_GLOBAL | S0_Taster_STOPP | X0 | %IX0 | BOOL | ... | FALSE | Öffnerkontakt |
| 1 | VAR_GLOBAL | S1_Schalter_TOR_AUF | X1 | %IX1 | BOOL | ... | FALSE | |
| 2 | VAR_GLOBAL | S2_Taster_TOR_AUF | X2 | %IX2 | BOOL | ... | FALSE | |
| 3 | VAR_GLOBAL | S3_Endschalter_Oben | X3 | %IX3 | BOOL | ... | FALSE | Öffnerkontakt |
| 4 | VAR_GLOBAL | S4_Taster_TOR_ZU | X4 | %IX4 | BOOL | ... | FALSE | Innen |
| 5 | VAR_GLOBAL | S5_Taster_TOR_ZU | X5 | %IX5 | BOOL | ... | FALSE | Aussen |
| 6 | VAR_GLOBAL | S6_Endschalter_Unten | X6 | %IX6 | BOOL | ... | FALSE | Öffnerkontakt |
| 7 | VAR_GLOBAL | S7_Lichtschranke | X7 | %IX7 | BOOL | ... | FALSE | |
| 8 | VAR_GLOBAL | H1_Warnleuchte | Y10 | %QX16 | BOOL | ... | FALSE | |
| 9 | VAR_GLOBAL | K1_Tor_Oeffnen | Y11 | %QX17 | BOOL | ... | FALSE | |
| 10 | VAR_GLOBAL | K2_Tor_Schliessen | Y12 | %QX18 | BOOL | ... | FALSE | |

### Eingabe des Programms

Nun werden die einzelnen Teilaufgaben der Steuerungsaufgabe gelöst:

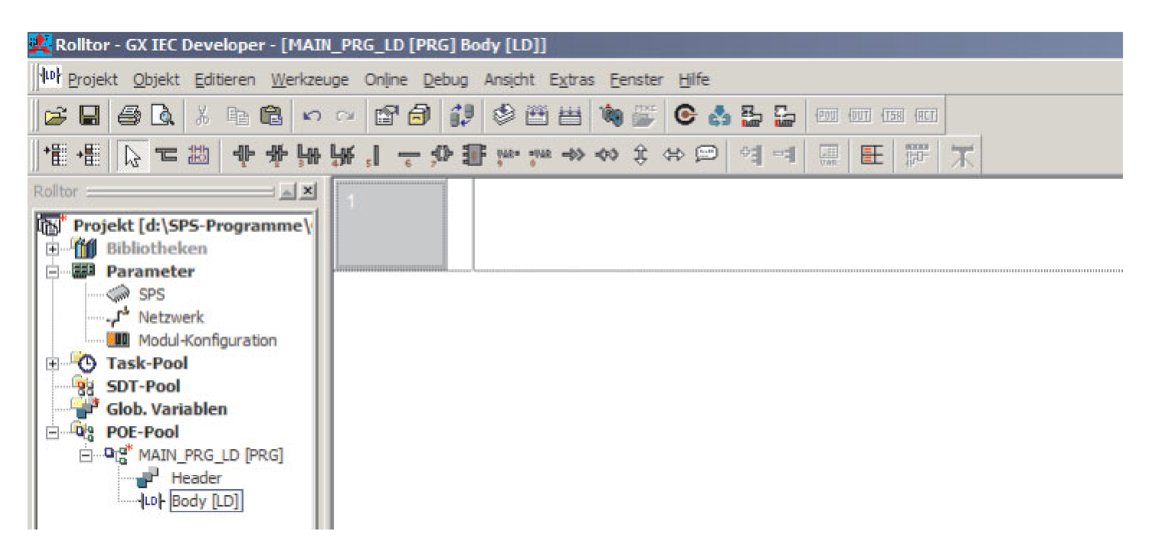
- Bedienung des Rolltores durch die Taster

Die Eingangssignale zur Bedienung des Tores müssen im Programm in zwei Befehle für den Antriebsmotor umgesetzt werden: „Tor öffnen“ und „Tor schließen“. Da es sich um Signale von Tastern handelt, die nur kurzzeitig an den Eingängen zur Verfügung stehen, müssen diese Signale gespeichert werden. Dazu werden zwei Variable gesetzt und zurückgesetzt, die im Programm zunächst stellvertretend für die Ausgänge stehen:

– TOR_OEFFNEN

– TOR_SCHLIESSEN

Falls der „Body“ der POE MAIN noch nicht angezeigt wird, klicken Sie im Navigatorfenster doppelt auf den Eintrag ***Body [LD]***.

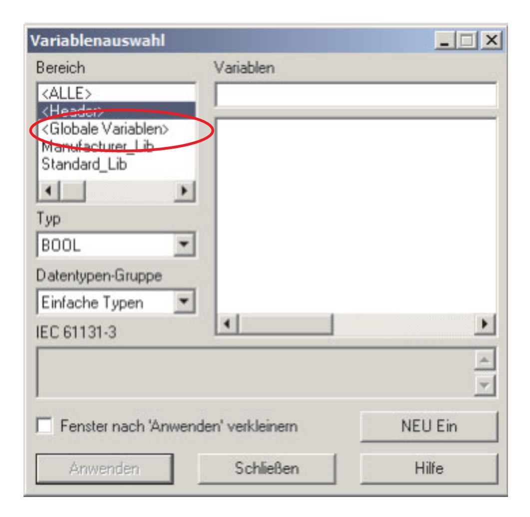





Klicken Sie in der Werkzeugleiste auf das Symbol „Kontakt“.

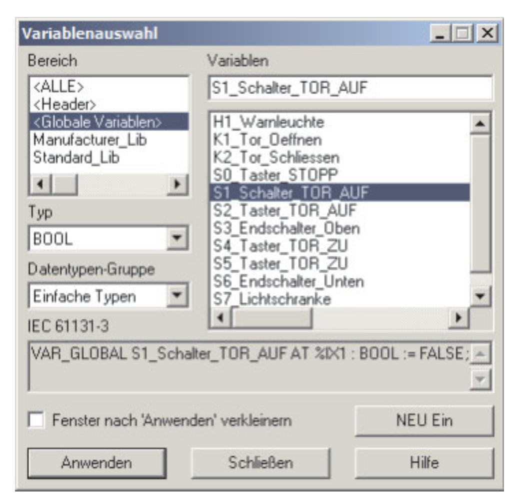


Setzen Sie den Cursor auf die gewünschte Stelle, und betätigen Sie die linke Maustaste.

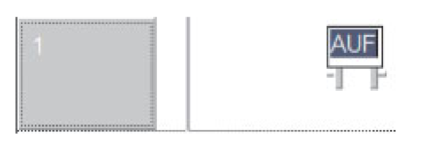



Durch einen Klick mit der rechten Maustaste auf das Fragezeichen öffnen Sie das Auswahlfenster für die Variablen.

Klicken Sie im Feld ***Bereich*** auf ***<Globale Variablen>***



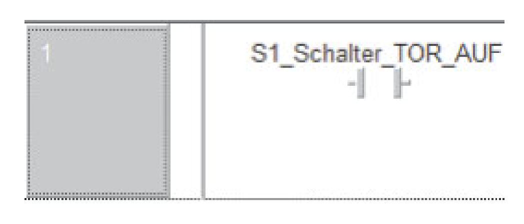

Markieren Sie die gewünschte Variable (in diesem Fall „S1_Schalter_TOR-AUF“) durch einen Mausklick.
Die gewählte Variable wird mit dem Schaltfeld ***Anwenden*** oder durch einen Doppelklick mit der linken Maustaste übernommen.




Die Variable wird eingefügt...




... und nach einem Mausklick in den Editierbereich auch mit dem gesamten Bezeichner angezeigt.

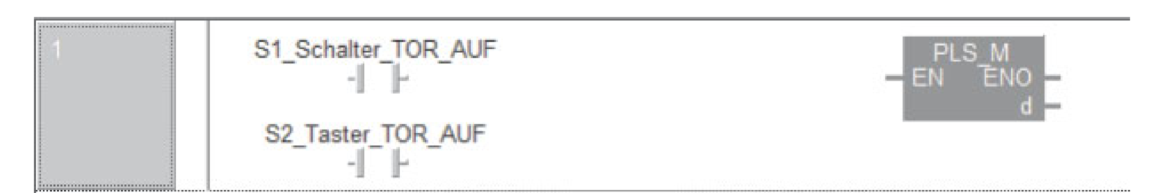



Positionieren Sie den Cursor auf den unteren Rand des Netzwerks, bis aus dem Cursor ein Doppelpfeil wird. Bewegen Sie dann bei betätigter linker Maustaste den Cursor nach unten, um das Netzwerk zu vergrößern.

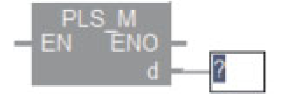



Geben Sie auch den anderen Schalter zum Öffnen des Tors ein.

Die Betätigung der „TOR-AUF“-Schalters muss in einen Impuls umgesetzt werden. Dazu wird die Funktion PLS_M verwendet. Wie eine Funktion eingegeben wird, wurde bereits im Anschnitt 4.7.7 beschrieben.




Klicken Sie in der Werkzeugleiste auf das Schaltfeld „Ausgangsvariable“ .



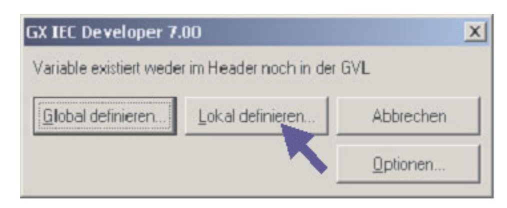

Klicken Sie anschließend auf den Ausgang der PLS_M-Funktion. Dadurch kann an dieser Stelle eine Ausgangsvariable eingegeben werden.

– Deklaration von lokalen Variablen

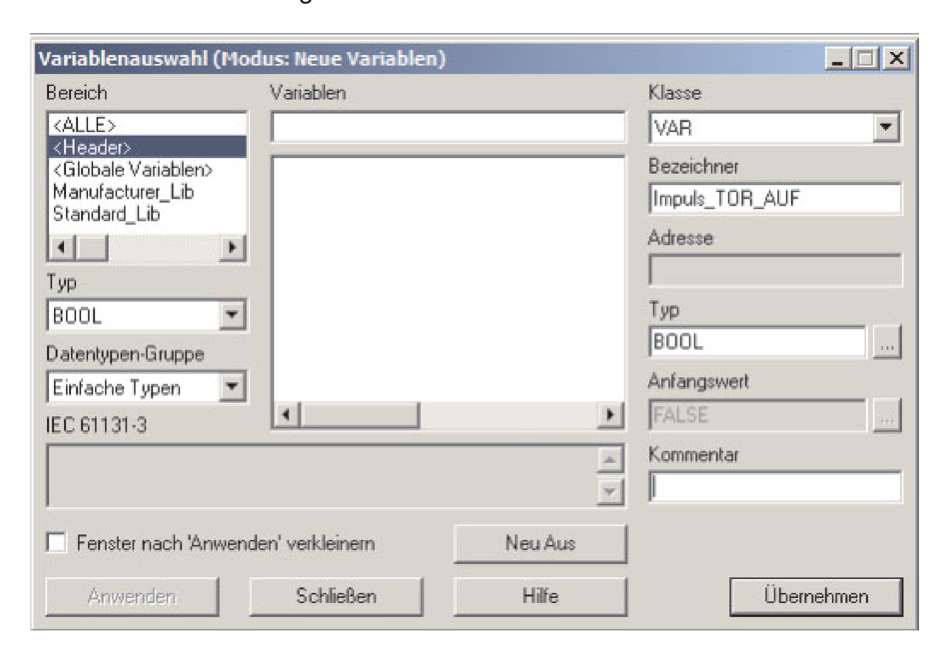
Diese Ausgangsvariable soll nur in dieser POE den Impuls weitergeben und kann daher eine lokale Variable sein. Bis jetzt wurden für dieses Projekt noch keine lokalen Variablen festgelegt, weil das auch während der Programmierung gemacht werden kann:

Tragen Sie in das leere Feld den Namen der Variablen ein: Impuls_TOR_AUF.

Da diese Variable noch nicht im deklariert wurde, wird das folgende Dialogfenster angezeigt:

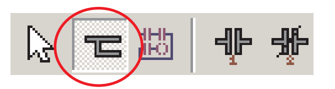



Klicken Sie auf ***Lokal definieren***. Danach wird das unten abgebildete Fenster zur Eingabe einer neuen Variablen geöffnet.



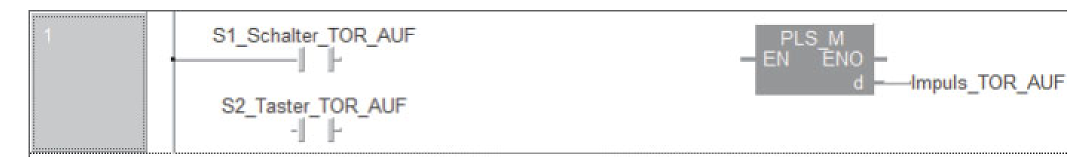

Klicken Sie auf ***Übernehmen***, um die neue Variable in die Liste der Lokalen Variablen (Header der POE) einzutragen.

Nun kann das Netzwerk fertiggestellt werden. Dazu müssen die einzelnen Elemente verbunden werden.



In der Werkzeugleiste finden Sie dazu das Symbol „Linie“. Bitte beachten Sie, dass der Cursor im Verbindungsmodus die Form eines Stiftes annimmt.

Positionieren Sie den Cursor auf die linke Sammelschiene, betätigen Sie die linke Maustaste, und ziehen Sie bei weiter betätigter linker Maustaste eine Linie bis zum Kontakt. Lassen sie hier die linke Maustaste los.

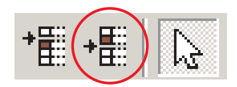



Verbinden Sie auch die anderen Elemente dieses Netzwerks.

1
S1_Schalter_TOR_AUF
S2_Taster_TOR_AUF
PLS_M
EN ENO
d
Impuls_TOR_AUF

– Einfügen eines neuen Programmnetzwerks

Um unter dem momentan bearbeiteten Netzwerk ein neues Netzwerk einzufügen, klicken Sie in der Werkzeugleiste bitte auf dieses Schaltfeld:



Es erscheint ein leeres Netzwerk:

1
S1_Schalter_TOR_AUF
S2_Taster_TOR_AUF
PLS_M
EN ENO
d
Impuls_TOR_AUF
2

In dieses und in weitere Netzwerke werden die folgenden Programmelemente eingegeben:

1
S1_Schalter_TOR_AUF
S2_Taster_TOR_AUF
PLS_M
EN ENO
d
Impuls_TOR_AUF
2
Impuls_TOR_AUF TOR_SCHLIESSEN TOR_OEFFNEN
S
3
S4_Taster_TOR_ZU
S5_Taster_TOR_ZU
PLS_M
EN ENO
d
Impuls_TOR_SCHLIESSEN
4
Impuls_TOR_SCHLIESSEN TOR_OEFFNEN TOR_SCHLIESSEN
S

Alle Variablen außer den Tastern und Schaltern sind lokale Variablen. Hier zeigt sich bereits ein Vorteil bei der Verwendung von Variablen mit symbolischen Namen: Auch ohne Eingabe von Operandenkommentaren ist das Programm übersichtlicher als bei der Verwendung absoluter Adressen wie X1, X2 usw.

- Funktionsbeschreibung zu den Netzwerken 1 bis 4

Zuerst werden die Signale zum Öffnen des Tores verarbeitet: Wird der Schlüsseltaster S1 oder der Taster S2 betätigt, wird ein Impuls gebildet, der nur einen Programmzyklus den Signalzustand „1“ hat. Dadurch kann das Tor nicht durch Festhalten oder Festklemmen eines Tasters blockiert werden. Die Auswertung der Taster S4 und S5 zum Schließen des Tores ist ähnlich realisiert. Der Antrieb darf nur eingeschaltet werden, wenn er sich nicht in die Gegenrichtung dreht. Aus diesem Grund kann das Tor nur geöffnet werden, wenn das Tor nicht gerade geschlossen wird und umgekehrt.

**HINWEIS** Die Verriegelung der Drehrichtungen muss außerhalb der SPS noch durch eine Verriegelung durch die Schützkontakte ergänzt werden (siehe Schaltplan im Abschnitt 4.9.3.)

- Tor nach 20 Sekunden automatisch schließen

5 S3_Endschalter_Oben TIMER_M EN ENO TC0 TCoil 200 TValue
6 TS0 TOR_SCHLIESSEN S

Wenn das Tor geöffnet ist, wird S3 betätigt und der Eingang X3 ausgeschaltet. (S3 hat aus Sicherheitsgründen einen Öffnerkontakt.) Nun beginnt die mit Timer T0 realisierte Verzögerungszeit von 20 s (200 x 0,1 s = 20 s). Nach Ablauf dieser Zeit wird die lokale Variable „TOR_SCHLIESSEN" gesetzt und dadurch das Tor geschlossen.

**HINWEIS** Zeitgeber (Timer) werden im nächsten Kapitel ausführlich erklärt.

- Tor mittels STOP-Schalter anhalten

7 S0_Taster_STOPP TOR_OEFFNEN R TOR_SCHLIESSEN R

Durch Betätigung des STOP-Tasters S0 werden die beiden lokalen Variablen zurückgesetzt und dadurch das Tor angehalten.

- Hinderniserkennung durch Lichtschranke

8 S7_Lichtschranke TOR_SCHLIESSEN TOR_SCHLIESSEN R TOR_OEFFNEN S

Wenn die Lichtschranke während des Schließens ein Hindernis erkennt, wird der Schließvorgang beendet und das Tor wieder geöffnet.

- Abschaltung des Motors durch Endschalter

9 S3_Endschalter_Oben TOR_OEFFNEN R
10 S6_Endschalter_Unten TOR_SCHLIESSEN R

Durch das geöffnete Tor wird der Endschalter S3 betätigt und der Eingang X3 ausgeschaltet. Dies setzt die lokale Variable TOR_OEFFNEN zurück und stoppt dadurch den Antrieb.

Erreicht das Tor die untere Position, wird S6 betätigt, X6 ausgeschaltet und ebenfalls der Antrieb angehalten. Aus Sicherheitsgründen haben die Endschalter Öffnerkontakte. Dadurch wird der Antrieb auch bei einer Unterbrechung der Verbindung zwischen Schalter und Eingang ausgeschaltet bzw. das Einschalten verhindert.

**HINWEIS** Die Endschalter müssen auch unabhängig von der SPS den Antrieb stoppen und in die Verdrahtung mit einbezogen werden (siehe Schaltplan im Abschnitt 4.9.3).

- Motorsteuerung

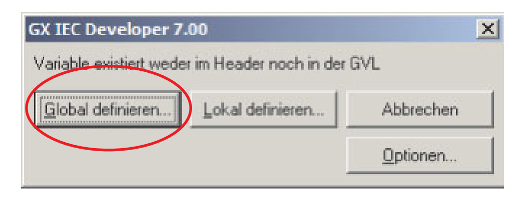



Am Ende des Programms werden die Signalzustände der beiden lokalen Variablen für Öffnen/Schließen auf die Ausgänge Y11 bzw. Y12 übertragen.

- Warnleuchte: „Tor in Bewegung“ und „Tor in undefinierter Position“

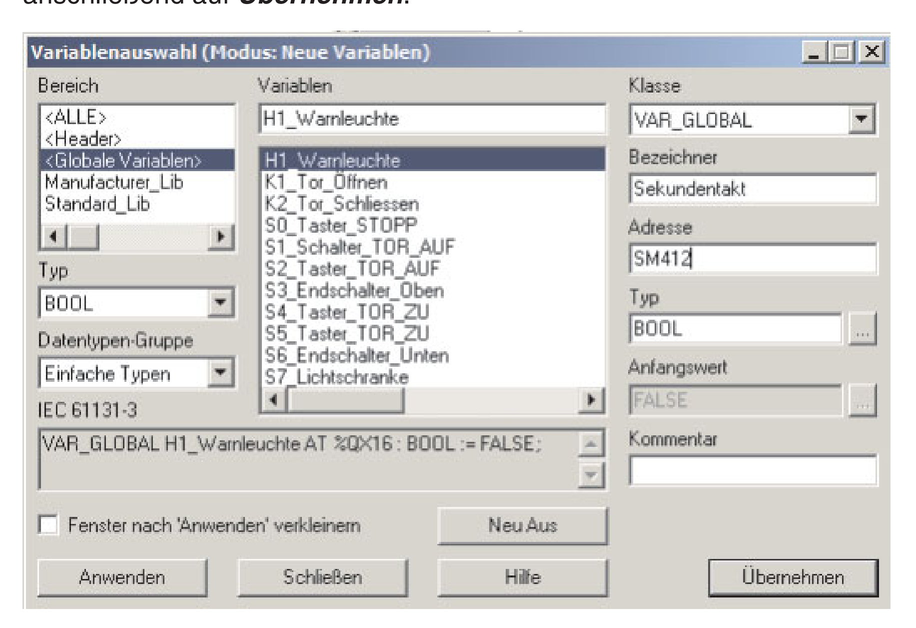


Wenn keiner der beiden Endschalter betätigt ist, wird das Tor entweder geöffnet oder geschlossen oder es wurde in einer Zwischenstellung angehalten. In diesen Fällen blinkt eine Warnleuchte. Als Blinktakt wird der Sondermerker SM412 verwendet, der automatisch im 1-s-Takt gesetzt und zurückgesetzt wird (siehe Abschnitt 5.2). SM412 wird während der Programmeingabe als Globale Variable definiert:

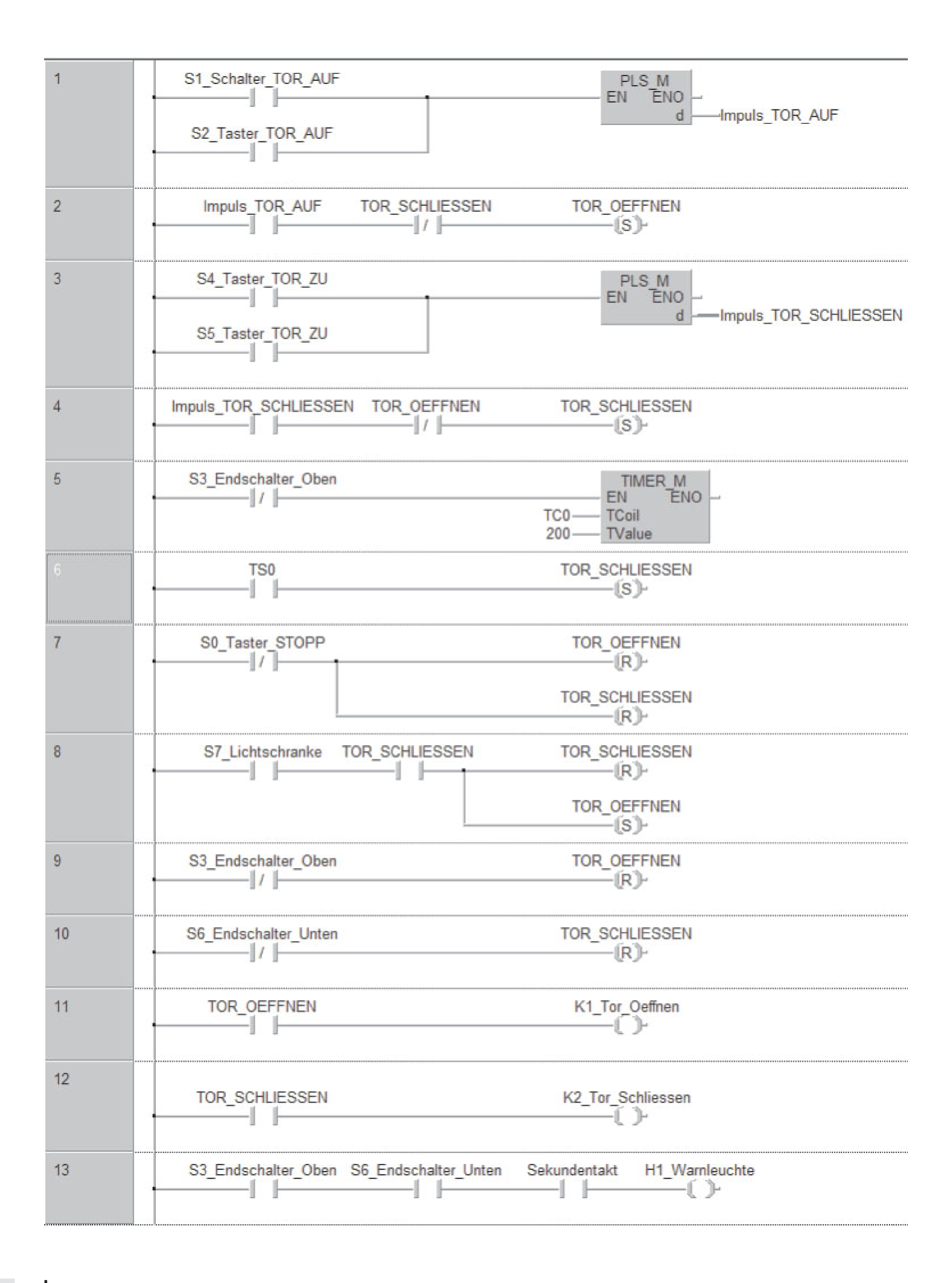



Geben Sie den Namen der Variablen (z. B. Sekundentakt) ein. Da es diese Variable noch nicht gibt, erscheint die links abgebildete Meldung. Klicken Sie auf ***Global definieren***.

Im Dialogfenster ***Variablenauswahl*** geben Sie im Feld Adresse SM412 ein und klicken anschließend auf ***Übernehmen***.




Die Abbildung auf der folgenden Seite zeigt noch einmal das gesamte Kontaktplanprogramm zur Steuerung des Rolltors.




**HINWEIS**

Ganz wichtig ist die Reihenfolge der Anweisungen und besonders das Zurücksetzen der Variablen TOR_OEFFNEN und TOR_SCHLIESSEN durch die Sicherheitseinrichtungen am Ende der Programmsequenz **nach** dem Setzen dieser Variablen.
Durch die Ausführung der Anweisungen „von oben nach unten“ (Abschnitt 2.2) hat so das Ausschalten und damit die Sicherheit immer Vorrang vor dem Einschalten.

### 4.9.3 Die Hardware

Zur Steuerung des Rolltors werden in diesem Beispiel die folgenden Komponenten aus dem MELSEC System Q verwendet:

- Hauptbaugruppenträger mit mindestens zwei Steckplätzen für E/A-Module, z. B. Q33B
- Netzteil Q62P

  Dieses Netzteil stellt eine Gleichspannung von 24 V zur Verfügung, die für die Versorgung der Schalter und Meldegeräte verwendet werden kann. Beachten Sie aber bitte, das dieser Ausgang maximal 0,6 A liefern kann.
- CPU-Modul (nach Bedarf)*
- 1 digitales Eingangsmodul QX80 mit 16 Eingängen (plusschaltende Geber)
- 1 digitales Ausgangsmodul QY80 mit 16 Transistorausgängen (plusschaltend)

* In der Praxis wird vermutlich niemand eine SPS des MELSEC System Q nur für die Steuerung eines Rolltores einsetzen. Die CPU wäre mit dieser Aufgabe kaum ausgelastet. Als Teil einer komplexen Anwendung, wie z. B. bei der Steuerung von Fertigungsprozessen, ist diese Anwendung aber durchaus denkbar.

**Anschluss der SPS**

S0 S1 S2 S3 S4 S5 S6 S7

L1 N PE

| L | N | FG | X00 | X01 | X02 | X03 | X04 | X05 | X06 | X07 | X08 | X09 | X0A | X0B | X0C | X0D | X0E | X0F | COM | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Netzteil | | | Digitales Eingangsmodul | | | | | | | | | | | | | | | | | |
| | | | Digitales Ausgangsmodul | | | | | | | | | | | | | | | | | |
| | +24V | 24G | Y10 | Y11 | Y12 | Y13 | Y14 | Y15 | Y16 | Y17 | Y18 | Y19 | Y1A | Y1B | Y1C | Y1D | Y1E | Y1F | COM | 0V |

K2 K1 Verriegelung durch Schützkontakte

S3 S6 Abschaltung durch Endschalter

H1 K1 K2

Eine Auflistung der elektrischen Betriebsmittel finden Sie auf der nächsten Seite.

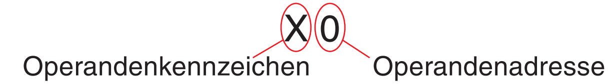
| Kenn-zeichen | Funktion | Adresse | Bemerkung |
|---|---|---|---|
| S0 | Tastschalter STOP | X0 | Öffnerkontakt |
| S1 | Schlüsselschalter Tor AUF (außen) | X1 | Schließerkontakte |
| S2 | Taster Tor AUF (innen) | X2 | |
| S3 | Endschalter oben (TOR geöffnet) | X3 | Öffnerkontakt |
| S4 | Taster Tor ZU (innen) | X4 | Schließerkontakte |
| S5 | Taster Tor ZU (außen) | X5 | |
| S6 | Endschalter unten (Tor ZU) | X6 | Öffnerkontakt |
| S7 | Lichtschranke | X7 | X7 wird „1“, wenn ein Hindernis erkannt wird |
| H1 | Warnleuchte | Y10 | — |
| K1 | Motorschütz (Motor Linkslauf) | Y11 | Linkslauf = Tor öffnen |
| K2 | Motorschütz (Motor Rechtslauf) | Y12 | Rechtslauf = Tor schließen |

# 5 Operanden im Detail

Die Operanden einer SPS werden in Steuerungsanweisungen verwendet, das heißt, ihre Signalzustände oder Werte können durch das SPS-Programm abgefragt oder beeinflusst werden. Ein Operand besteht aus

– einem Operandenkennzeichen und
– einer Operandenadresse.

Beispiel für die Angabe eines Operanden (z. B. Eingang 0):



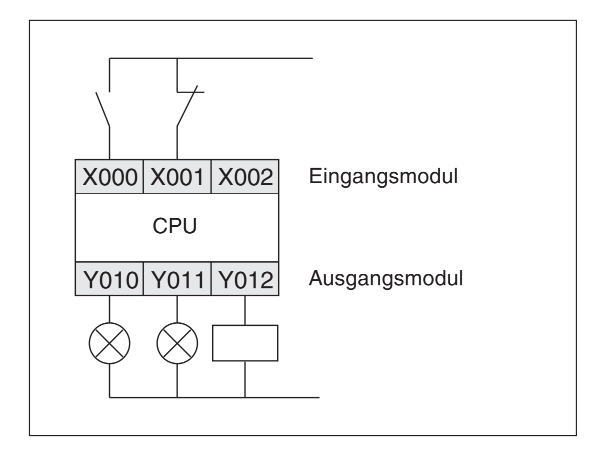

## 5.1 Ein- und Ausgänge

Die Ein- und Ausgänge verbinden eine SPS mit dem zu steuernden Prozess. Bei der Abfrage eines Eingangs durch das SPS-Programm wird die Spannung an einer Eingangsklemme eines Eingangsmoduls geprüft. Da es sich um digitale Eingänge handelt, können sie nur zwei Signalzustände annehmen: EIN und AUS. Wenn die Spannung an der Eingangsklemme einen definierten Wert (z.B. 24 V) erreicht, ist der Eingang eingeschaltet (Signalzustand „1“). Bei einer geringeren Spannung gilt der Eingang als ausgeschaltet (Signalzustand „0“).

Als Operandenkennzeichen für Eingänge wird bei einer MELSEC SPS „**X**“ verwendet. Derselbe Eingang kann beliebig oft im Programm abgefragt werden.

**HINWEIS** Durch das SPS-Programm kann der Zustand von Eingängen nicht verändert werden. Zum Beispiel ist die Angabe eines Eingangs als Operand einer OUT-Anweisung nicht möglich.

Wird als Operand einer Ausgabeanweisung ein Ausgang verwendet, wird das Verknüpfungsergebnis (der Signalzustand des Operanden) am Anschluss des Ausgangsmoduls ausgegeben. Bei Relaisausgängen zieht das entsprechende Relais an (alle Relais haben Schließerkontakte), und bei Steuerungen mit Transistorausgängen schaltet der angesprochene Transistor durch und damit den angeschlossenen Verbraucher ein.




Beispiel für den Anschluss von Schaltern an den Eingängen und Lampen oder Schützen an den Ausgängen einer MELSEC SPS.

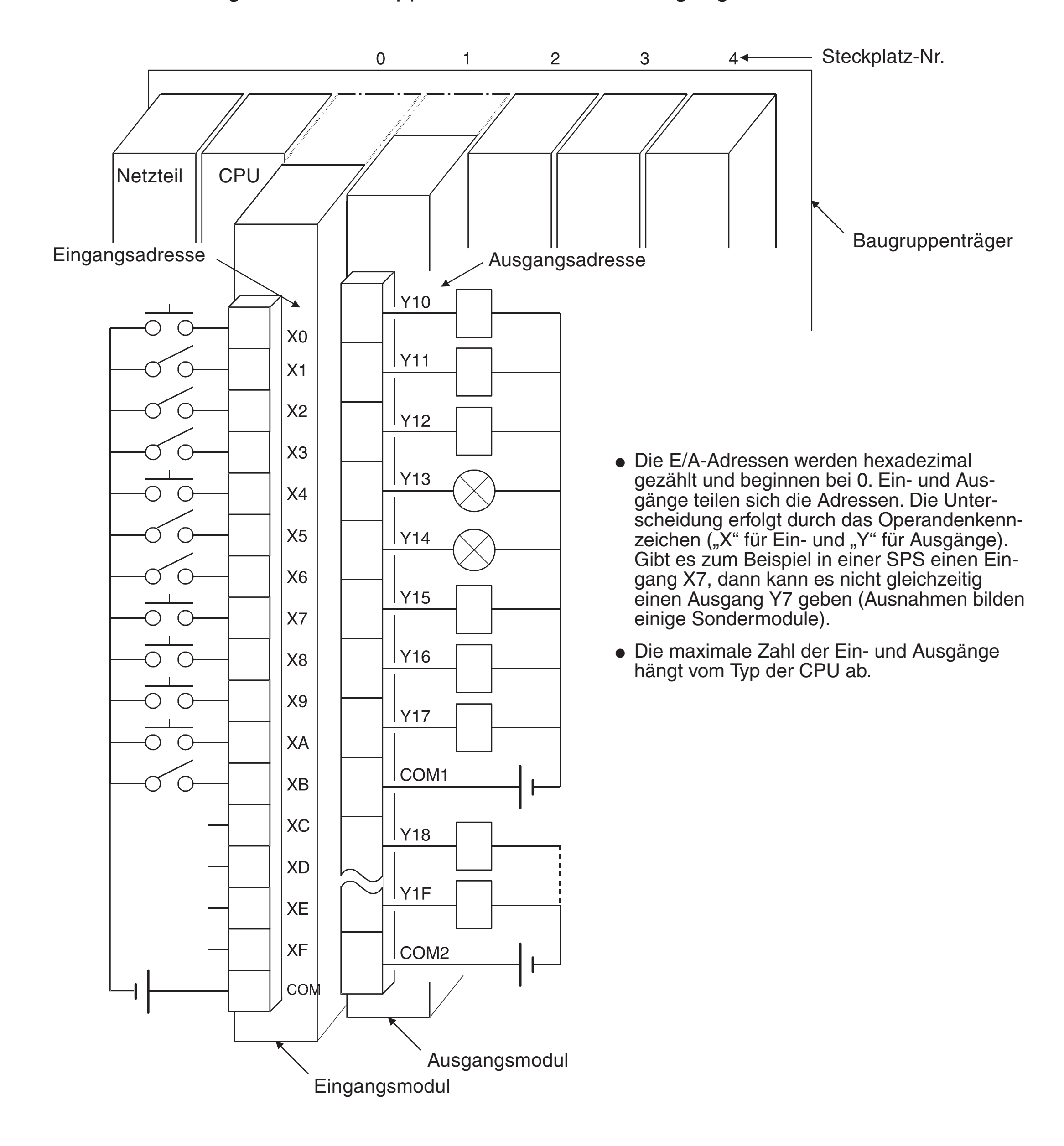
Das Operandenkennzeichen der Ausgänge ist „**Y**“. Ausgänge können nicht nur in Ausgabeanweisungen, sondern auch in Verknüpfungsanweisungen verwendet werden. Auf keinen Fall darf aber ein und derselbe Ausgang mehrmals als Operand einer Ausgabeanweisung programmiert werden (siehe auch Abschnitt 4.7.2).

## 5.1.1 Adressierung von Ein- und Ausgängen

Die Signale, die externe Geräte an die Eingänge der SPS liefern, werden für die Programmierung in Eingangsadressen umgewandelt. Die Adresse eines SPS-Eingangs wird dadurch bestimmt, auf welchem Steckplatz des Baugruppenträgers das Eingangsmodul installiert ist (siehe Abschnitt 3.2.2) und an welchem Eingang eines Moduls ein Signal angeschlossen ist.

Die Adressen der durch das Programm gesteuerten Ausgänge werden ebenfalls durch den Steckplatz und dem Anschluss am Modul bestimmt. Um ein externes Gerät zu schalten, muss dessen Anschluss mit dem entsprechenden SPS-Ausgang verbunden werden.

Die Ein- und Ausgänge werden hexadezimal (0, 1, 2 ...9, A, B, C, D, E, F; 10, 11, 12 ...) adressiert. Dadurch ergeben sich Gruppen zu 16 Ein- oder Ausgängen.




- Die E/A-Adressen werden hexadezimal gezählt und beginnen bei 0. Ein- und Ausgänge teilen sich die Adressen. Die Unterscheidung erfolgt durch das Operandenkennzeichen („X“ für Ein- und „Y“ für Ausgänge). Gibt es zum Beispiel in einer SPS einen Eingang X7, dann kann es nicht gleichzeitig einen Ausgang Y7 geben (Ausnahmen bilden einige Sondermodule).
- Die maximale Zahl der Ein- und Ausgänge hängt vom Typ der CPU ab.

## 5.1.2 Ein- und Ausgänge im MELSEC System Q

Die folgende Tabelle gibt eine Übersicht über die Ein- und Ausgänge der Steuerungen der SPS-CPUs des MELSEC System Q.

<table>
<tr><th colspan="2" rowspan="2">Operand</th><th colspan="2">Ein- und Ausgänge</th></tr>
<tr><th>E/A in Haupt- und Erweiterungsbau-gruppenträger</th><th>E/A in Haupt- und Erweiterungsbau-gruppenträger und dezentrale E/A</th></tr>
<tr><td colspan="2">Operandenkennzeichen</td><td colspan="2">X (Eingänge), Y (Ausgänge)</td></tr>
<tr><td colspan="2">Operandentyp</td><td colspan="2">Bit-Operand</td></tr>
<tr><td colspan="2">Werte, die ein Operand annehmen kann</td><td colspan="2">0 oder 1</td></tr>
<tr><td colspan="2">Angabe der Operanden-adresse</td><td colspan="2">Hexadezimal</td></tr>
<tr><td rowspan="10">Anzahl der Operanden und Adressen (abhängig vom Typ der CPU)</td><td>Q00J</td><td>256 (X/Y000 bis X/Y00FF)</td><td>2048 (X/Y000 bis X/Y07FF)</td></tr>
<tr><td>Q00</td><td rowspan="2">1024 (X/Y000 bis X/Y03FF)</td><td rowspan="2">2048 (X/Y000 bis X/Y07FF)</td></tr>
<tr><td>Q01</td></tr>
<tr><td>Q02</td><td rowspan="7">4096 (X/Y000 bis X/Y0FFF)</td><td rowspan="7">8192 (X/Y000 bis X/Y1FFF)</td></tr>
<tr><td>Q02H</td></tr>
<tr><td>Q06H</td></tr>
<tr><td>Q12H</td></tr>
<tr><td>Q25H</td></tr>
<tr><td>Q12PH</td></tr>
<tr><td>Q25PH</td></tr>
</table>

## 5.2 Merker

Oft müssen in einem SPS-Programm binäre Zwischenergebnisse (Signalzustand „0“ oder „1“) gespeichert werden. Für diesen Zweck stehen in einer SPS „Merker“ zur Verfügung (Operandenkennzeichen: „**M**“).

In Merkern wird das Verknüpfungs(zwischen)ergebnis zum Beispiel mit einer OUT-Anweisung eingetragen und kann danach mit Verknüpfungsanweisungen abgefragt werden. Merker helfen, das Programm übersichtlich zu gestalten und sparen Programmschritte. Verknüpfungsergebnisse, die mehrmals im Programm benötigt werden, können auf einen Merker gelegt und danach beliebig oft verwendet werden.

M1

M1

Abfrage auf Signalzustand „1“ (Merker gesetzt?)

M1

Abfrage auf Signalzustand „0“
(Ist der Merker zurückgesetzt?)

Die MELSEC-Steuerungen besitzen neben „normalen“ Merkern auch sogenannte Latch-Merker (Operandenkennzeichen: „**L**“). Die normalen, ungepufferten Merker werden beim Ausschalten der Versorgungsspannung der SPS auf den Signalzustand „0“ zurückgesetzt und haben diesen Zustand auch nach dem Einschalten der SPS. Latch-Merker dagegen behalten auch bei einem Spannungsausfall ihre Informationen.

<table>
<tr><th colspan="2" rowspan="2">Operand</th><th colspan="2">Merker</th></tr>
<tr><th>Ungepufferte Merker</th><th>Latch-Merker</th></tr>
<tr><td colspan="2">Operandenkennzeichen</td><td>M</td><td>L</td></tr>
<tr><td colspan="2">Operandentyp</td><td colspan="2">Bit-Operand</td></tr>
<tr><td colspan="2">Werte, die ein Operand annehmen kann</td><td colspan="2">0 oder 1</td></tr>
<tr><td colspan="2">Angabe der Operandenadresse</td><td colspan="2">Dezimal</td></tr>
<tr><td rowspan="10">Anzahl der Operanden und Adressen</td><td>Q00J</td><td rowspan="10">8192 (M0–M8191)*</td><td rowspan="10">8192 (L0–L8191)*</td></tr>
<tr><td>Q00</td></tr>
<tr><td>Q01</td></tr>
<tr><td>Q02</td></tr>
<tr><td>Q02H</td></tr>
<tr><td>Q06H</td></tr>
<tr><td>Q12H</td></tr>
<tr><td>Q25H</td></tr>
<tr><td>Q12PH</td></tr>
<tr><td>Q25PH</td></tr>
</table>

* Die Anzahl der Merker und Latch-Merker kann in den SPS-Parametern verändert werden. Die hier angegebenen Werte entsprechen der Voreinstellung.

### 5.2.1 Sondermerker

Neben den Merkern, die vom Anwender im Programm beliebig ein- und ausgeschaltet werden können, existieren noch *Sondermerker* mit dem Operandenkennzeichen „**SM**“. Diese Merker zeigen bestimmte Systemzustände an oder beeinflussen die Programmbearbeitung. Die folgende Tabelle zeigt nur eine kleine Auswahl der Sondermerker.

| Sonder-merker | Beschreibung | Verarbeitung im Programm |
|---|---|---|
| SM0 | SPS-Fehler | Abfrage des Signalzustands |
| SM51 | Niedrige Batteriespannung | |
| SM400 | In der Betriebsart „RUN“ der SPS ist der Signalzustand dieses Merkers immer „1“. | |
| SM401 | In der Betriebsart „RUN“ der SPS ist der Signalzustand dieses Merkers immer „0“. | |
| SM402 | Initialisierungsimpuls (Nach dem Einschalten der Betriebsart „RUN“ ist dieser Merker für die Dauer eines Programmzyklus „1“.) | |
| SM411 | Taktgeber, Periodendauer 0, 2 Sekunden (0,1 s EIN, 0,1 s AUS) | |
| SM412 | Taktgeber, Periodendauer 1 Sekunde (0,5 s EIN, 0,5 s AUS) | |
| SM413 | Taktgeber, Periodendauer 2 Sekunden (1 s EIN, 1 s AUS) | |
| SM414 | Einstellbarer Takt | |

**HINWEIS**

Eine Übersicht aller Sondermerker enthält die Programmieranleitung zur A/Q-Serie (Art.-Nr. 87 432).

## 5.3 Timer

Bei der Steuerung von Prozessen oder Abläufen müssen oft bestimmte Vorgänge zeitverzögert ein- oder ausgeschaltet werden. Während hierfür in der Relaistechnik Zeitrelais verwendet werden, stehen in einer SPS interne Zeitglieder (engl.: *Timer*) zur Verfügung.

Im Prinzip zählen Timer einen internen Takt der SPS (z. B. Impulse im 0,1 s-Takt). Wenn der Zählwert einen über das Programm vorgegebenen Wert erreicht, wird der Ausgang des Timers eingeschaltet.

Zu einem Timer gehören vier Elemente:

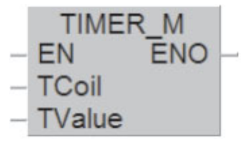
– Sollwert (***TValue***)
– Istwert (***TN***)
– Spule (***TCoil***, ***TC***)
– Ausgangskontakt (***TS***)

Alle Timer arbeiten als Einschaltverzögerung und werden durch Ansteuern der „Spule“ mit einem „1“-Signal aktiviert. Zum Starten und Rücksetzen der Timer stehen besondere OUT-Anweisungen zur Verfügung. Der Ausgang eines Timers (TS) kann innerhalb des Programms beliebig oft abgefragt werden.

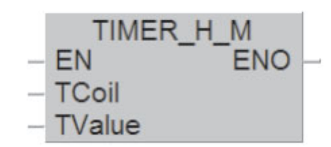
Beim MELSEC System Q werden langsame und schnelle Timer unterschieden. Mit Hilfe der Programmier-Software kann in den SPS-Parametern die Zeitbasis (das ist der Takt, mit dem der Timer zählt) für langsame Timer im Bereich von 1 ms bis 1000 ms eingestellt werden. Voreingestellt sind 100 ms. Für schnelle Timer kann eine Zeitbasis von 0,1 ms bis 100 ms gewählt werden (Voreinstellung: 10 ms).

Ob ein Timer als schneller oder langsamer Timer arbeiten soll, wird durch die Anweisung festgelegt, die den Timer startet.

Aufruf eines langsamen Timers

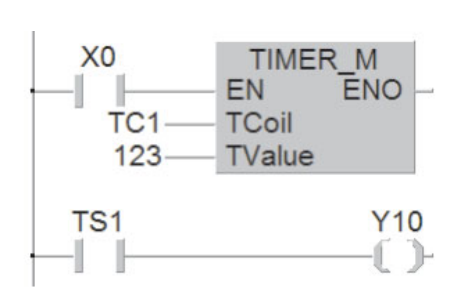



Aufruf eines schnellen Timers




**Beispiel zur Programmierung eines langsamen Timers**

Kontaktplan




Am Eingang TCoil der TIMER_M-Anweisung wird die Operandenadresse des Timers angegeben (in diesem Beispiel **TC1**).

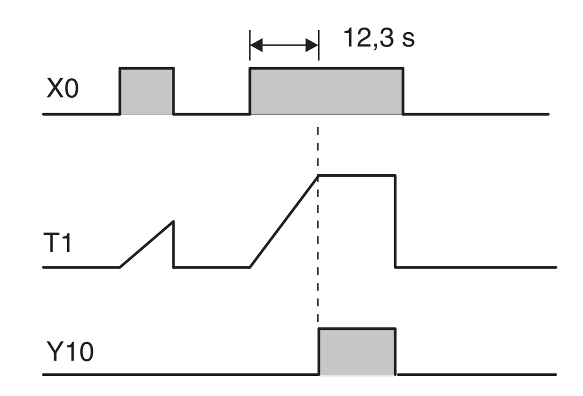
MELSEC-Anweisungsliste

| | |
|---|---|
| LD | X0 |
| OUT | T1 |
| | K123 |
| LD | T1 |
| OUT | Y10 |

IEC-Anweisungsliste

| | | |
|---|---|---|
| LD | X0 | |
| TIMER_M | TC1, | 123 |
| LD | TS1 | |
| ST | Y10 | |

Der Timer T1 wird gestartet, wenn der Eingang X0 eingeschaltet ist. Der Sollwert beträgt 123 x 100 ms = 12,3 s. Nach Ablauf dieser Zeit schaltet T1 den Ausgang Y10 ein. Für das oben dargestellte Beispiel ergibt sich der folgende Signalverlauf:




Solange X0 eingeschaltet ist, zählt der Timer die internen 100 ms-Impulse. Beim Erreichen des Sollwerts wird der Ausgang von T1 eingeschaltet.

Wird der Eingang X0 oder die Versorgungsspannung der SPS ausgeschaltet, wird der Timer zurückgesetzt und dessen Ausgang ebenfalls ausgeschaltet.

Die Angabe des Zeitsollwertes kann auch durch den Inhalt eines Datenregisters vorgenommen werden. Diese Möglichkeit ist im Abschnitt 5.7.1 beschrieben.

**Remanente Zeitglieder**

Die CPUs des MELSEC System Q verfügen neben den oben beschriebenen Zeitgliedern auch über remanente Zeitglieder, die auch nach dem Abschalten der ansteuernden Verknüpfung den bereits erreichten Zeitistwert behalten. Die Zeitistwerte werden in einem Speicher abgelegt, dessen Inhalt auch bei einem Spannungsausfall erhalten bleibt.

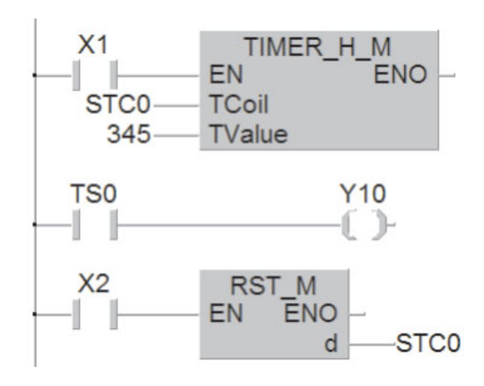
Das Operandenkennzeichen der remanenten Timer ist „ST“. Wie die „normalen“ Timer können auch remanente Timer als schnelle oder langsame Timer programmiert werden.

**HINWEIS**

Bei Auslieferung einer CPU sind in den SPS-Parametern 2048 (2k) normale Timer und keine remanenten Timer eingestellt. Um remanente Timer programmieren zu können, muss deren Anzahl in den SPS-Parametern festgelegt werden.

Beispiel für die Programmierung eines remanenten Timers als schnellenTimer:

Kontaktplan



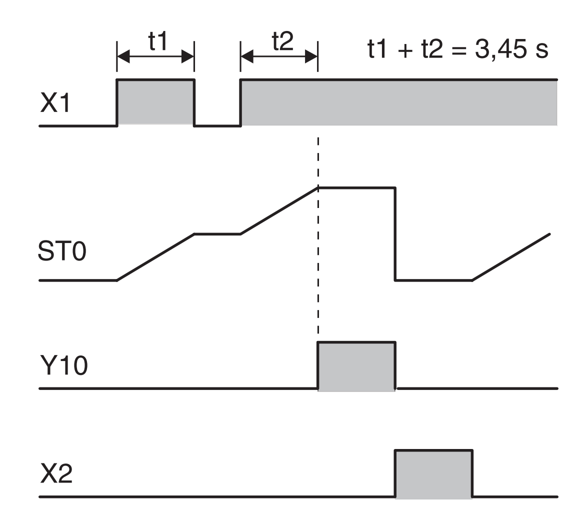

MELSEC-Anweisungsliste

| | |
|---|---|
| LD | X1 |
| OUTH | ST0 |
| | K345 |
| LD | ST0 |
| OUT | Y10 |
| LD | X2 |
| RST | ST0 |

IEC-Anweisungsliste

| | |
|---|---|
| LD | X1 |
| TIMER_H_M | STC0, 345 |
| LD | STS0 |
| OUT | Y10 |
| LD | X2 |
| R | STC0 |

Der Timer ST0 wird gestartet, wenn der Eingang X1 eingeschaltet ist. Als Sollwert ist 345 x 10 ms = 3,45 s vorgegeben. Nach Erreichen des Sollwerts schaltet ST0 den Ausgang Y10 ein. Mit dem Eingang X2 wird der Timer zurückgesetzt und dessen Ausgang ausgeschaltet.




Wenn X1 eingeschaltet ist, zählt der Timer die internen 10 ms-Impulse. Auch wenn X1 ausgeschaltet wird, bleibt der bis dahin erreichte Istwert erhalten. Wenn der Istwert dem Sollwert entspricht, wird der Ausgang des Timers eingeschaltet.

Weil beim Ausschalten des Eingangs X1 oder der Versorgungsspannung der SPS der Zeitistwert nicht gelöscht wird, ist dafür eine besondere Anweisung im Programm notwendig. Mit dem Eingang X2 wird der Timer ST0 zurückgesetzt und dessen Ausgang ausgeschaltet.

### Übersicht der Timer bei den SPS-CPUs des MELSEC System Q

| Operand | | Timer | |
|---|---|---|---|
| | | Normale Timer | Remanente Timer |
| Operandenkennzeichen | | T | ST |
| Operandentyp (zur Ansteuerung und Abfrage) | | Bit-Operand | |
| Werte, die ein Operand (Timer-Ausgang) annehmen kann | | 0 oder 1 | |
| Angabe der Operandenadresse | | Dezimal | |
| Vorgabe des Zeitsollwerts | | Als ganzzahlige, dezimale Konstante. Die Vorgabe erfolgt entweder direkt in der Anweisung oder durch Speicherung in einem Datenregister. | |
| Anzahl der Operanden und Adressen | Q00J | 512 (T0 bis T511)* | 0* |
| | Q00 | | |
| | Q01 | | |
| | Q02 | 2048 (T0 bis T2047)* | 0* |
| | Q02H | | |
| | Q06H | | |
| | Q12H | | |
| | Q25H | | |
| | Q12PH | | |
| | Q25PH | | |

* Voreinstellung, die Anzahl der Timer kann in den SPS-Parametern verändert werden.

## 5.4 Zähler (Counter)

Zur Programmierung von Zählvorgängen stehen Ihnen bei den Steuerungen des MELSEC System Q interne Zähler (engl.: *Counter*) zur Verfügung.

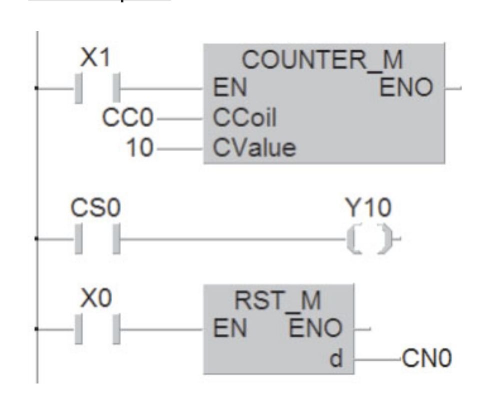
Counter zählen Signale, die ihrem Eingang per Programm zugeführt werden. Erreicht der Zählwert einen ebenfalls über das Programm vorgegebenen Sollwert, wird der Ausgang des Counters eingeschaltet. Dieser kann innerhalb des Programms beliebig oft abgefragt werden.

Ein Counter besitzt vier Elemente:

– Sollwert (***CValue***)
– Istwert (***CN***)
– Spule (***CCoil, CC***)
– Ausgangskontakt (***CS***)

**Beispiel für die Programmierung eines Counters:**

Kontaktplan




Am Eingang CCoil der COUNTER_M-Anweisung wird die Operandenadresse des Counters angegeben (in diesem Beispiel C0).

MELSEC-Anweisungsliste

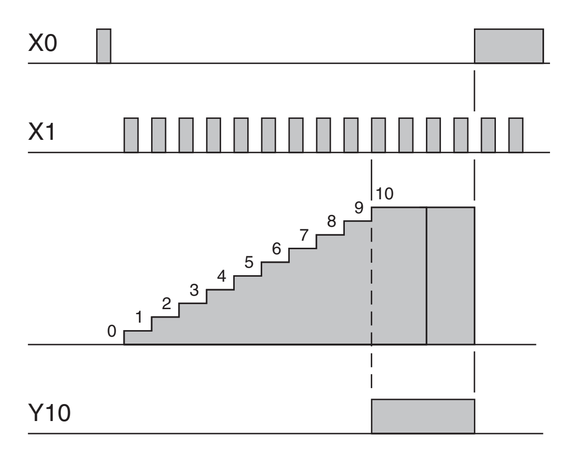
| | |
|---|---|
| LD | X1 |
| OUT | C0 |
| | K10 |
| LD | C0 |
| OUT | Y10 |
| LD | X0 |
| RST | C0 |

IEC-Anweisungsliste

| | |
|---|---|
| LD | X1 |
| COUNTER_M | CC0, 10 |
| LD | CS0 |
| ST | Y10 |
| LD | X0 |
| R | CN0 |

Jedesmal, wenn der Eingang X1 eingeschaltet wird, zählt der Counter C0 um den Zahlenwert 1 aufwärts. Der Ausgang Y10 wird gesetzt, nachdem der Eingang X1 zehn mal ein- und ausgeschaltet wurde (Als Zählersollwert ist „10“ programmiert).

Die folgende Abbildung zeigt den Signalverlauf für dieses Programmbeispiel.




Durch den Eingang X0 wird der Zähler mit Hilfe einer RST-Anweisung zurückgesetzt. Der Istwert des Counters wird auf 0 gesetzt und der Ausgang des Counters ausgeschaltet.

Nach Erreichen des Zählersollwertes wird der Counter von den dann folgenden Impulsen am Eingang X1 nicht mehr beeinflusst.

Die folgende Tabelle zeigt die wichtigsten Merkmale der Counter.

| Merkmal | Counter |
|---|---|
| Wirkungsweise | Bei einer steigenden Flanke des Signals am Counter-Eingang wird der Zählwert um 1 erhöht. (Es ist nicht notwendig, den Zähleingang mit einen Impuls anzusteuern.) |
| Zählrichtung | Aufwärtszählend |
| Bereich für Sollwert | 1 bis 32767 |
| Vorgabe des Sollwerts | Als dezimale Konstante direkt in der Anweisung oder gespeichert in einem Datenregister. |
| Verhalten bei Zählerüberlauf | Zählt maximal bis 32767, danach ändert sich der Istwert nicht mehr |
| Zählerausgang | Nach Erreichen des Sollwerts bleibt der Ausgang eingeschaltet. |
| Zurücksetzen | Mit einer RST-Anweisung wird der Istwert des Zählers gelöscht und der Ausgang ausgeschaltet. |

## Übersicht der Counter

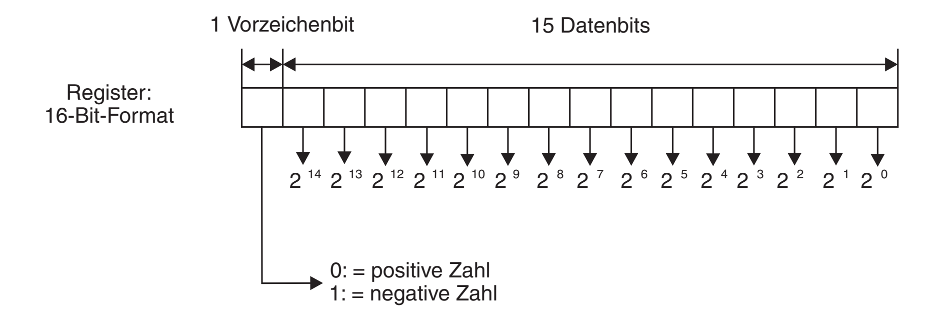
| Operand | | Counter |
|---|---|---|
| Operandenkennzeichen | | C |
| Operandentyp (zur Ansteuerung und Abfrage) | | Bit-Operand |
| Werte, die der Counter-Ausgang annehmen kann | | 0 oder 1 |
| Angabe der Operandenadresse | | Dezimal |
| Vorgabe des Zählersollwerts | | Als dezimale Konstante direkt in der Anweisung oder gespeichert in einem Datenregister. |
| Anzahl der Operanden und Adressen | Q00J | 512* (C0 bis C511) |
| | Q00 | |
| | Q01 | |
| | Q02 | 1024* (C0 bis C1023) |
| | Q02H | |
| | Q06H | |
| | Q12H | |
| | Q25H | |
| | Q12PH | |
| | Q25PH | |

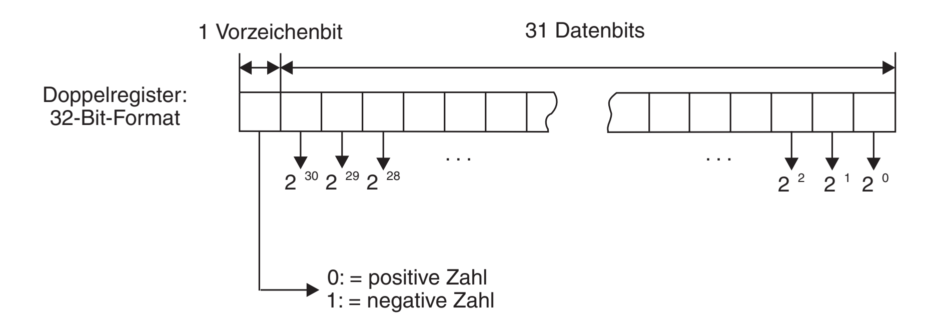
* Voreinstellung, die Anzahl der Counter kann in den SPS-Parametern verändert werden.

# 5.5 Register

In einer SPS dienen Merker zum Speichern von binären Zwischenergebnissen. Der Zustand eines Merkers liefert aber nur eine Ein/Aus- oder 0/1-Information und ist daher nicht zur Speicherung von Messwerten oder Ergebnissen von Berechnungen geeignet. Für diesen Zweck sind die Steuerungen des MELSEC System Q mit Registern ausgestattet.

Ein Register besteht aus 16 Bit oder einem Wort (siehe Abschnitt 4.2). Durch Zusammenschalten von zwei 16-Bit-Registern kann ein „Doppelregister“ mit 32 Bit gebildet werden.




In einem Register können Werte im Bereich von 0000H bis FFFFH (-32768 bis 32767) gespeichert werden, während ein Doppelregister Werte im Bereich von 00000000H bis FFFFFFFFH (-2 147 483 648 bis 2 147 483 647) enthalten kann.

Für den Umgang mit Registern bieten die CPUs des MELSEC System Q eine große Anzahl von Anweisungen, mit denen z. B. Werte in Register geschrieben, aus Registern gelesen, die Inhalte von Registern kopiert, verglichen oder in arithmetischen Berechnungen verarbeitet werden können (siehe Kap. 6).

## 5.5.1 Datenregister

Datenregister können im SPS-Programm als Speicher verwendet werden. Ein Wert, der durch das SPS-Programm in ein Datenregister eingetragen wird, bleibt dort unverändert erhalten, bis er im Programm durch einen anderen Wert überschrieben wird.

Bei Anweisungen für 32-Bit-Daten wird nur die Adresse eines 16-Bit-Registers angegeben, das folgende Register wird automatisch mit dem höherwertigen Teil der 32-Bit-Daten belegt. Wenn z. B. zur Speicherung eines 32-Bit-Wertes das Register D0 angegeben wird, enthält D0 die Bits 0 bis 15 und D1 die Bits 16 bis 31 des Wertes.

### Verhalten beim Ausschalten oder Stoppen der SPS

In den SPS-Parametern können Datenregisterbereiche festgelegt werden (sog. Latch-Bereiche), deren Inhalt bei einem Stopp der SPS oder beim Ausschalten der Versorgungsspannung der SPS nicht gelöscht wird.

### Datenregister im Überblick

| Operand | | Datenregister |
|---|---|---|
| Operandenkennzeichen | | D |
| Operandentyp | | Wort-Operand (Zwei Register können zu einem Doppelregister zusammengefasst werden.) |
| Werte, die ein Operand annehmen kann | | 16-Bit-Register: 0000H bis FFFFH (-32768 bis 32767)<br>32-Bit-Register: 00000000H bis FFFFFFFFH (-2 147 483 648 bis 2 147 483 647) |
| Angabe der Operandenadresse | | Dezimal |
| Anzahl der Operanden und Adressen | Q00J | 11136* (D0 bis D11135) |
| | Q00 | |
| | Q01 | |
| | Q02 | 12288* (D0 bis D12287) |
| | Q02H | |
| | Q06H | |
| | Q12H | |
| | Q25H | |
| | Q12PH | |
| | Q25PH | |

* Voreinstellung, die Anzahl der Datenregister kann in den SPS-Parametern verändert werden.

## 5.5.2 Sonderregister

Neben den Sondermerker (Abschnitt 5.2.1) sind die CPU-Module des MELSEC System Q mit Sonderregistern ausgestattet. Das Operandenkennzeichen dieser Register ist „**SD**“. Oft besteht sogar ein direkter Zusammenhang zwischen Sondermerker und Sonderregister. So zeigt beispielsweise der Sondermerker SM51 an, dass die Spannung einer Batterie der SPS zu niedrig ist und der Inhalt des Sonderregisters SD51 gibt an, welche Batterie betroffen ist (CPU oder Speicherkarte). Eine kleine Auswahl der Sonderregister ist in der folgenden Tabelle aufgeführt.

| Sonderegister | Beschreibung | Verarbeitung im Programm |
|---|---|---|
| SD0 | Fehlercode | Abfrage des Inhalts |
| SD392 | Software-Version | |
| SD520, SD521 | Aktuelle Programmzykluszeit | |
| SD210–SD213 | Uhrzeit und Datum der integrierten Uhr (BCD-Format) | Abfrage des Inhalts<br>Verändern des Inhalts |
| SD414 | Periodendauer des Taktes von SM414 | Verändern des Inhalts |

**HINWEIS** Eine Übersicht aller Sonderregister enthält die Programmieranleitung zur A/Q-Serie und zum MELSEC System Q (Art.-Nr. 87 432).

### 5.5.3 File-Register

Der Inhalt von File-Registern geht auch beim Auschalten der Versorgungsspannung nicht verloren. Aus diesem Grund können in File-Registern Werte gespeichert werden, die nach dem Einschalten der SPS in Datenregister übertragen werden und die das Programm z. B. für Berechnungen, Vergleiche oder als Sollwerte für Timer benötigt.

File-Register unterscheiden sich im Aufbau nicht von Datenregistern.

<table>
<tr><th colspan="2">Operand</th><th>File-Register</th></tr>
<tr><td colspan="2">Operandenkennzeichen</td><td>R</td></tr>
<tr><td colspan="2">Operandentyp</td><td>Wort-Operand (Zwei Register können zu einem Doppelregister zusammengefasst werden.)</td></tr>
<tr><td colspan="2">Werte, die ein Operand annehmen kann</td><td>16-Bit-Register: $0000_H$ bis $FFFF_H$ (-32768 bis 32767)<br>32-Bit-Register: $00000000_H$ bis $FFFFFFFF_H$ (-2 147 483 648 bis 2 147 483 647)</td></tr>
<tr><td colspan="2">Angabe der Operandenadresse</td><td>Dezimal</td></tr>
<tr><td rowspan="10">Anzahl der Operanden und Adressen</td><td>Q00J</td><td>0</td></tr>
<tr><td>Q00</td><td rowspan="2">32767 (R0 bis R32766)</td></tr>
<tr><td>Q01</td></tr>
<tr><td>Q02</td><td rowspan="7">32767 pro Block (R0 bis R32766)<br>Durch Verwendung einer Speicherkarte können bis zu 1 Million zusätzliche File-Register gespeichert werden.</td></tr>
<tr><td>Q02H</td></tr>
<tr><td>Q06H</td></tr>
<tr><td>Q12H</td></tr>
<tr><td>Q25H</td></tr>
<tr><td>Q12PH</td></tr>
<tr><td>Q25PH</td></tr>
</table>

# 5.6 Konstanten

## 5.6.1 Dezimal- und Hexadezimalkonstanten

Mit den Dezimal- und Hexadezimalkonstanten lassen sich nummerische Zahlenwerte innerhalb eines SPS-Programms festlegen (z. B. Zeit- oder Zählersollwerte). Der Zahlenwert wird intern von der SPS in einen binären Zahlenwert codiert.

Bei der Programmierung im Kontaktplan oder der IEC-Anweisungliste werden dezimale Konstanten nicht besonders gekennzeichnet. Bei hexadezimalen Konstanten werden dem Zahlenwert die Zeichen „16#“ vorangestellt. Beispielsweise interpretiert die SPS-CPU die Angabe „16#12“ als hexadezimalen Wert 12.

In der MELSEC-Anweisungsliste wird der Konstanten der Buchstabe „K“ oder „H“ vorangestellt. Beispiele: K100 (Dezimaler Wert „100“), H64 (Hexadezimaler Wert „64“)

Die folgende Tabelle zeigt die Wertebereiche der Dezimal- und Hexadezimalkonstanten.

| Konstanten | 16 Bit | 32 Bit |
|---|---|---|
| **Dezimal** | -32 768 bis +32 767 | -2 147 483 648 bis +2 147 483 647 |
| **Hexadezimal** | 0 bis FFFF | 0 bis FFFFFFFF |

## 5.6.2 Konstanten mit Gleitkommazahlen

Dezimale Konstanten sind ganze Zahlen ohne Nachkommastellen. Gleitkommazahlen dagegen können Stellen vor und nach einem Dezimalkomma aufweisen und bieten dadurch Vorteile bei arithmetischen Operationen.

Im Programm werden Konstante, die aus Gleitkommazahlen bestehen, durch ein vorangestelltes „E“ gekennzeichnet (Zum Beispiel E1.234 oder E1.234 + 3). Gleitkommazahlen können auf zwei verschiedene Arten definiert werden:

– Angabe einer Konstanten ohne Exponenten

  Der Wert wird wie üblich angegeben. Das Komma muss allerdings durch einen Punkt ersetzt werden. Zum Beispiel kann der Wert „10,2345“ im Programm als „E10.2345“ übergeben werden.

– Angabe einer Konstanten mit Exponenten

  Der Wert wird mit Basis und einem Exponenten angegeben. Der Exponent hat die Basis 10 ($10^n$). Der Wert „1234“ zum Beispiel kann auch als „1,234 x 1000“ oder – in expontieller Schreibweise – als „1,234 x $10^3$“ dargestellt werden. Im Programm wird dieser Wert als „E1.234 + 3“ angegeben („+3“ entspricht „$10^3$“).

Gleitkommazahlen können Werte aus den folgenden Bereichen annehmen:
-1,0 x $2^{128}$ bis -1,0 x $2^{-126}$,
0
und 1,0 x $2^{-126}$ bis 1,0 x $2^{+128}$

## 5.6.3 Konstante Zeichenfolgen

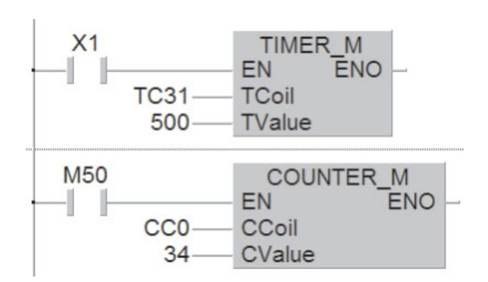
Werden Zeichen innerhalb eines Programms in Anführungsstrichen angegeben, werden sie als ASCII-Zeichen interpretiert (z. B. "MOTOR12"). Eine Zeichen belegt 1 Byte. Eine Zeichenfolge kann bis zu 32 Zeichen enthalten.

# 5.7 Programmiertipps für Timer und Counter

## 5.7.1 Indirekte Sollwertvorgabe bei Timern und Countern

Zeit- und Zählsollwerte können Timern und Countern direkt im Programm übergeben werden:

Kontaktplan




MELSEC-Anweisungsliste

| | |
|---|---|
| LD | X1 |
| OUT | T31 |
| | K500 |
| LD | M50 |
| OUT | C0 |
| | K34 |

IEC-Anweisungsliste

| | |
|---|---|
| LD | X1 |
| TIMER_M | TC31, 500 |
| LD | M50 |
| COUNTER_M | CC0, K34 |

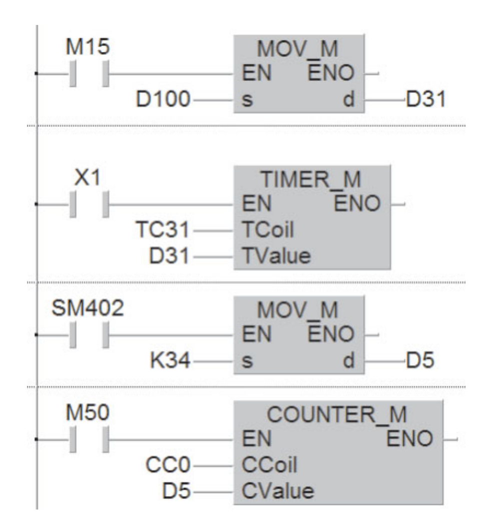
T31 im oben abgebildeten Beispiel ist ein 100 ms-Timer. Mit der Konstanten „500“ wird die Verzögerungszeit auf 500 x 0,1 s = 50 s eingestellt. Der Sollwert für den Counter C0 wird direkt auf „34“ eingestellt.

Der Vorteil dieser Art der Sollwertvorgabe liegt darin, dass man sich nicht weiter um den Sollwert kümmern muss. Auch nach einem Spannungsausfall oder direkt nach dem Einschalten gelten die durch das Programm vorgegebenen Sollwerte. Der Nachteil ist allerdings, dass für eine Sollwertänderung das Programm geändert werden muss. Besonders die Sollwerte von Timern werden oft erst während der Inbetriebnahme der Steuerung und dem Programmtest angepasst.

Die Sollwerte für Timer und Counter können aber auch in Datenregister eingetragen und durch das Programm aus diesen Registern gelesen werden. Dadurch können die Vorgabewerte schnell mit einem angeschlossenen Programmiergerät geändert werden. Auch die Vorgabe von Sollwerten durch Schalter in einem Pult oder einem Bediengerät ist in diesem Fall möglich.

Die Abbildung auf der nächsten Seite zeigt Beispiele für die indirekte Vorgabe von Sollwerten.

Kontaktplan




MELSEC-Anweisungsliste

| | |
|---|---|
| LD | M15 |
| MOV | D100 |
| | D31 |
| LD | X1 |
| OUT | T31 |
| | D131 |
| LD | SM402 |
| MOV | K34 |
| | D5 |
| LD | M50 |
| OUT | C0 |
| | D5 |

IEC-Anweisungsliste

| | |
|---|---|
| LD | M15 |
| MOV_M | D100, D31 |
| LD | X1 |
| TIMER_M | TC31, D31 |
| LD | SM402 |
| MOV_M | K34, D5 |
| LD | M50 |
| COUNTER_M | CC0, D5 |

- Wenn der Merker M15 „1“ ist, wird der Inhalt des Datenregisters D100 in das Datenregister D31 kopiert. Dieses Register enthält den Sollwert für T31. Der Inhalt von D100 kann z. B. durch ein Bediengerät verändert werden.
- Der Sondermerker SM402 ist nur nach dem Start der SPS für einen Programmzyklus eingeschaltet. Dadurch wird nach dem Einschalten der SPS die Konstante „34“ in das Datenregister D5 eingetragen, das als Sollwertspeicher für den Counter C0 dient.

Die Sollwerte müssen nicht unbedingt im SPS-Programm in die Datenregister eingetragen werden. Sie können auch vor dem Start des Programms mit Hilfe eines Programmiergeräts festgelegt werden.



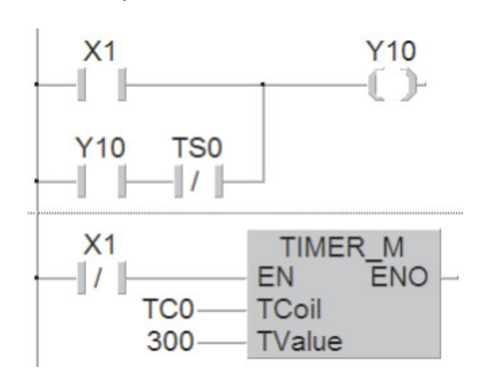
**ACHTUNG:**
***Verwenden Sie zur Speicherung von Sollwerten für Timer und Counter Latch-Datenregister, wenn die Sollwerte nicht durch das SPS-Programm in die Register eingetragen werden. Beachten Sie, das auch die Inhalte dieser Register verloren gehen, wenn die Pufferbatterie erschöpft ist.***
***Werden normale Register verwendet, werden die Sollwerte gelöscht, wenn die Versorgungsspannung ausgeschaltet oder der RUN/STOP-Schalter in die Stellung STOP geschaltet wird. Nach dem Einschalten der Spannung oder dem nächsten Start der SPS können durch die nun auf „0“ eingestellten Sollwerte gefährliche Zustände auftreten.***

## 5.7.2 Ausschaltverzögerung

Alle Timer einer MELSEC SPS arbeiten als Einschaltverzögerung. Der Timer-Ausgang wird nach dem Ablauf der vorgegebenen Zeit eingeschaltet. Oft werden aber Ausschaltverzögerungen benötigt. (Ein Anwendungsbeispiel ist die Steuerung eines Lüfters, der nach dem Ausschalten der Badezimmerbeleuchtung noch einige Minuten eingeschaltet bleibt.)

**Programmvariante 1 (Selbsthaltung)**

Kontaktplan




MELSEC-Anweisungsliste

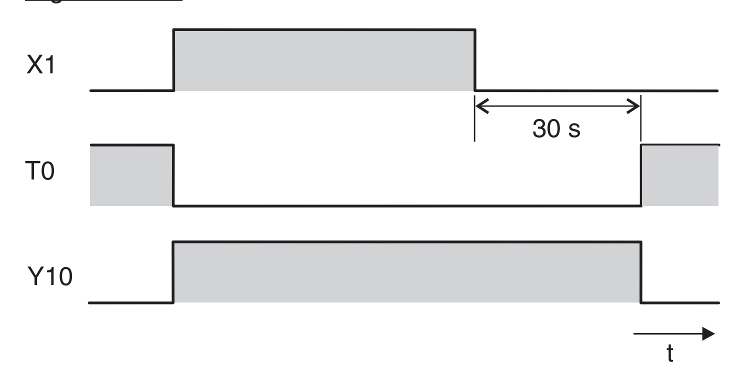
| | |
|---|---|
| LD | X1 |
| LD | Y10 |
| ANI | T0 |
| ORB | |
| OUT | Y10 |
| LDI | X1 |
| OUT | T0 |
| | K300 |

IEC-Anweisungsliste

| | |
|---|---|
| LD | X1 |
| OR( | |
| | Y10 |
| ANDN | TS0 |
| ) | |
| ST | Y10 |
| LDN | X1 |
| TIMER_M | TC0, 300 |

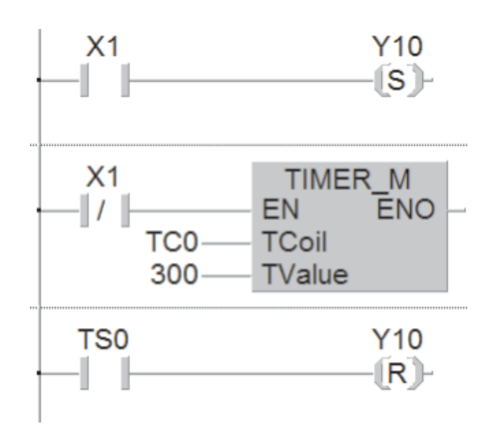
Solange der Eingang X1 (z. B. Lichtschalter) eingeschaltet ist, ist auch der Ausgang Y10 (Lüfter) eingeschaltet. Aber auch nach dem Ausschalten von X1 bleibt Y10 über die Selbsthaltung eingeschaltet, weil der Timer T0 noch nicht abgelaufen ist. Dieser wird mit dem Ausschalten von X1gestartet. Nach Ablauf der eingestellten Zeit (im Beispiel 300 x 0,1 s = 30 s) unterbricht T0 die Selbsthaltung von Y10 und dieser Ausgang wird ausgeschaltet.

Signalverlauf



**Programmvariante 2 (Setzen/Rücksetzen)**

Kontaktplan




MELSEC-Anweisungsliste

| | |
|---|---|
| LD | X1 |
| SET | Y10 |
| LDI | X1 |
| OUT | T0 |
| | K300 |
| LD | T0 |
| RST | Y000 |

IEC-Anweisungsliste

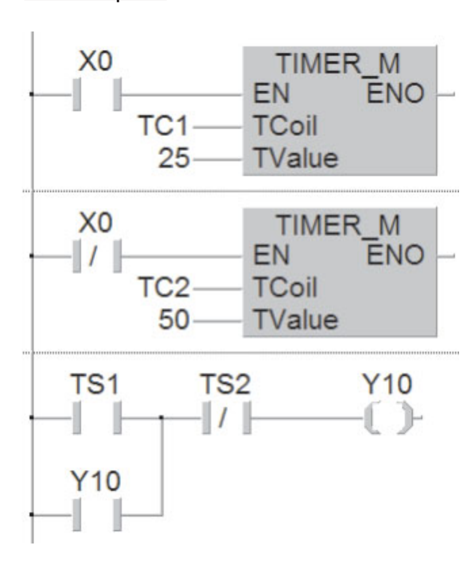
| | |
|---|---|
| LD | X1 |
| S | Y10 |
| LDN | X1 |
| TIMER_M | TC0, 300 |
| LD | TS0 |
| R | Y10 |

Beim Einschalten von X1 wird der Ausgang Y10 gesetzt (eingeschaltet). Beim Ausschalten von X1 wird T0 gestartet. Nach Ablauf der eingestellten Zeit setzt T0 den Ausgang Y10 zurück. Der Signalverlauf ist identisch mit der Programmvariante 1.

### 5.7.3 Ein- und Ausschaltverzögerung

In der Praxis kann es vorkommen, dass ein Ausgang verzögert eingeschaltet und gleichzeitig verzögert ausgeschaltet werden soll. Auch diese Aufgabe ist leicht mit den logischen Grundverknüpfungen zu lösen.

Kontaktplan



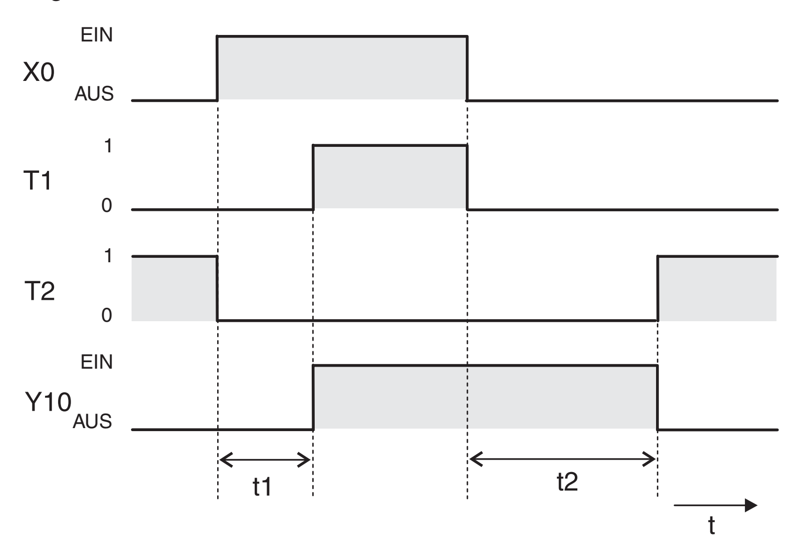

MELSEC-Anweisungsliste

| | |
|---|---|
| LD | X0 |
| OUT | T1 |
| | K25 |
| LDI | X0 |
| OUT | T2 |
| | K50 |
| LD | T1 |
| OR | Y10 |
| ANI | T2 |
| OUT | Y10 |

IEC-Anweisungsliste

| | |
|---|---|
| LD | X0 |
| TIMER_M | TC1, 25 |
| LDN | X0 |
| TIMER_M | TC2, 50 |
| LD | TS1 |
| OR | Y10 |
| ANDN | TS2 |
| ST | Y10 |

Signalverlauf




Beim Einschalten von X0 wird T1 gestartet und T2 zurückgesetzt. Erst nach Ablauf der Zeit t1 wird der Ausgang Y10 eingeschaltet und bleibt solange eingeschaltet, wie X0 EIN ist.

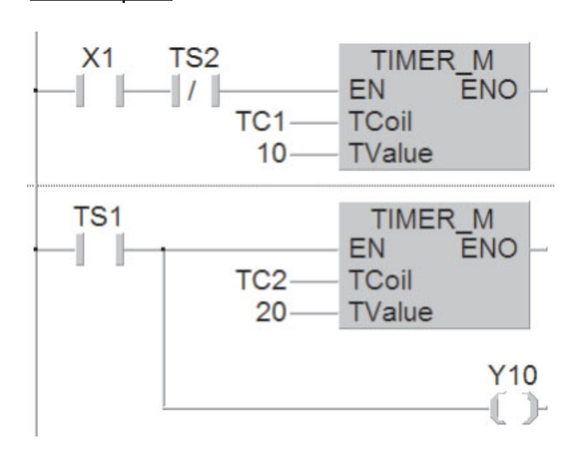
Auch wenn X0 ausgeschaltet und dadurch T1 zurückgesetzt wird, bleibt Y10 durch die Selbsthaltung über T1 zunächst eingeschaltet. Mit dem Ausschalten von X0 wird T2 gestartet. Dieser Timer schaltet nach Ablauf der Zeit t2 den Ausgang Y10 verzögert aus.

## 5.7.4 Taktgeber

In der SPS-CPU stehen Sondermerker zur Verfügung, mit denen sehr einfach Programmieraufgaben gelöst werden können, bei denen ein fester Takt verlangt wird (z. B. zur Ansteuerung einer Leuchte zur Störungsmeldung). SM413 z. B. wird im 1-Sekunden-Takt ein- und ausgeschaltet. Eine detaillierte Beschreibung aller Sondermerkern enthält die Programmieranleitung für die A/Q-Serie und das MELSEC System Q, Art.-Nr. 87432.

Wenn jedoch andere Taktzeiten oder unterschiedliche Ein- und Ausschaltzeiten verlangt werden, kann ein Taktgeber mit zwei Timern realisiert werden.

Kontaktplan




MELSEC-Anweisungsliste

| | |
|---|---|
| LD | X1 |
| ANI | T2 |
| OUT | T1 |
| | K10 |
| LD | T1 |
| OUT | T2 |
| | K20 |
| OUT | Y10 |

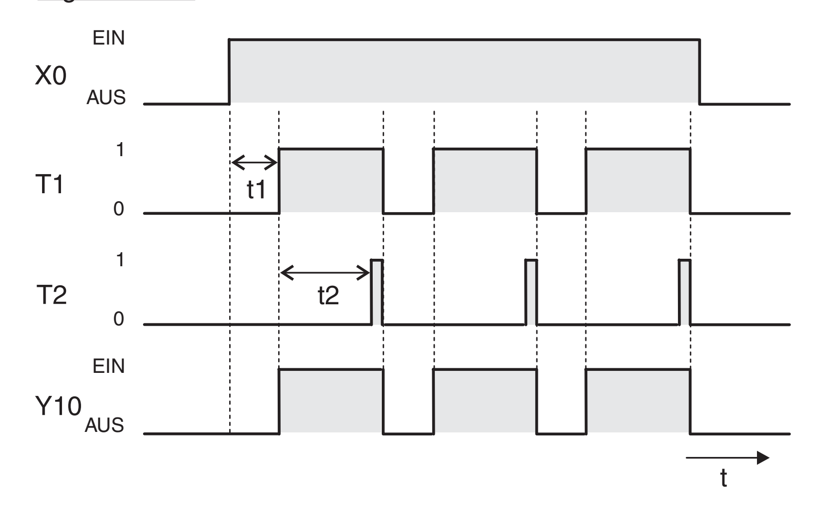
IEC-Anweisungsliste

| | |
|---|---|
| LD | X1 |
| ANDN | TS2 |
| TIMER_M | TC1, 10 |
| LD | TS1 |
| TIMER_M | TC2, 20 |
| ST | Y10 |

X1 startet den Taktgeber. Dieser Eingang kann auch weggelassen werden. Dann ist der Taktgeber ständig aktiviert. Im weiteren Programm wird der Ausgang von T1 z. B. für Meldeleuchten verarbeitet. Die Einschaltzeit wird durch T2 und die Ausschaltzeit durch T1 bestimmt.

Der Ausgang von Timer T2 wird nur für einen Programmzyklus eingeschaltet. In der folgenden Abbildung, die den Signalverlauf des Beispielprogramms zeigt, ist diese Zeit übertrieben lang dargestellt. T2 schaltet T1 aus und dadurch wird auch sofort danach T2 ausgeschaltet. Genau genommen verlängert sich also die Einschaltzeit um die Zeit, die für die Ausführung des Programms benötigt wird. Da die Zykluszeit aber nur einige Millisekundenbereich beträgt, kann sie in der Regel vernachlässigt werden.

Signalverlauf



# 6 Programmierung für Fortgeschrittene

Mit den in Kapitel 4 beschriebenen logischen Grundanweisungen kann eine speicherprogrammierbare Steuerung die Funktionen von Schützsteuerungen nachbilden. Damit sind aber die Möglichkeiten einer SPS noch lange nicht erschöpft. Da das Herzstück jeder SPS ein Mikroprozessor ist, stellen auch Berechnungen, Vergleiche von Zahlen, Umwandlungen von Zahlensystemen oder die Verarbeitung von analogen Werten kein Problem dar.

Zur Ausführung dieser Funktionen, die über die logischen Verknüpfungen hinausgehen, werden besondere Anweisungen, die sogenannten Applikationsanweisungen benötigt.

## 6.1 Übersicht der Applikationsanweisungen

Applikationsanweisungen sind durch eine Abkürzung, die sich aus der Beschreibung ihrer Funktion ableitet, eindeutig gekennzeichnet. Zum Beispiel lautet die Bezeichnung für die Anweisung, mit der 16-Bit-Daten transferiert werden können, „MOV". (Vom englischen *to move*, bewegen oder verschieben, da alle Abkürzungen für Applikationsanweisungen aus dem Englischen kommen.)

Die folgende Tabelle zeigt eine Übersicht aller Applikationsanweisungen, um die Möglichkeiten des MELSEC System Q zu verdeutlichen. Erschrecken Sie nicht, Sie müssen sich nicht alle Abkürzungen merken. Bei der Programmierung können Sie die Hilfe-Funktion der Programmier-Software GX Developer oder GX IEC Developer nutzen. Alle Anweisungen sind in der Programmieranleitung zur A/Q-Serie und zum MELSEC System Q, Artikel-Nr. 87 432, ausführlich und mit Beispielen beschrieben. In diesem Kapitel wird daher nur auf die am häufigsten verwendeten Anweisungen eingegangen (Diese sind in der Tabelle grau hinterlegt).

**HINWEIS**

Viele der Applikationsanweisungen können zyklisch oder auch nur bei steigender Flanke der Eingangsbedingung ausgeführt werden. In diesem Fall wird der Anweisung ein „P" nachgestellt.
Zum Beispiel: **MOV** -> zyklischer Datentransfer, solange die Eingangsbedingung erfüllt ist; **MOVP** -> einmaliger Datentransfer bei bei steigender Flanke der Eingangsbedingung.

| Einteilung | | Anweisung | Bedeutung |
|---|---|---|---|
| Vergleichsanweisungen | 16-Bit-Datenvergleich innerhalb von Verknüpfungen | **LD=** | Vergleich auf „Gleich" |
| | | **LD>** | Vergleich auf „Größer" |
| | | **LD<** | Vergleich auf „Kleiner" |
| | | **LD<>** | Vergleich auf „Ungleich" |
| | | **LD<=** | Vergleich auf „Kleiner-Gleich" |
| | | **LD>=** | Vergleich auf „Größer-Gleich" |
| | | **AND=** | Vergleich auf „Gleich" |
| | | **AND>** | Vergleich auf „Größer" |
| | | **AND<** | Vergleich auf „Kleiner" |
| | | **AND<>** | Vergleich auf „Ungleich" |
| | | **AND<=** | Vergleich auf „Kleiner-Gleich" |
| | | **AND>=** | Vergleich auf „Größer-Gleich" |
| | | **OR=** | Vergleich auf „Gleich" |
| | | **OR>** | Vergleich auf „Größer" |
| | | **OR<** | Vergleich auf „Kleiner" |
| | | **OR<>** | Vergleich auf „Ungleich" |
| | | **OR<=** | Vergleich auf „Kleiner-Gleich" |
| | | **OR>=** | Vergleich auf „Größer-Gleich" |

| Einteilung | | Anweisung | Bedeutung |
|---|---|---|---|
| Vergleichsan-weisungen | 32-Bit-Datenvergleich | LDD= | Vergleich von 32-Bit-Daten innerhalb von Verknüpfungen |
| | | LDD> | |
| | | LDD< | |
| | | LDD<> | |
| | | LDD<= | |
| | | LDD>= | |
| | | ANDD= | |
| | | ANDD> | |
| | | ANDD< | |
| | | ANDD<> | |
| | | ANDD>= | |
| | | ANDD<= | |
| | | ORD= | |
| | | ORD> | |
| | | ORD< | |
| | | ORD<> | |
| | | ORD<= | |
| | | ORD>= | |
| | Vergleich von Gleit-kommazahlen | LDE= | Vergleich innerhalb von Verknüpfungen |
| | | LDE> | |
| | | LDE< | |
| | | LDE<> | |
| | | LDE<= | |
| | | LDE>= | |
| | | ANDE= | |
| | | ANDE> | |
| | | ANDE< | |
| | | ANDE<> | |
| | | ANDE>= | |
| | | ANDE<= | |
| | | ORE= | |
| | | ORE> | |
| | | ORE< | |
| | | ORE<> | |
| | | ORE<= | |
| | | ORE>= | |
| | Vergleich von Zeichenfolgen | LD$= | Zeichenweiser Vergleich von Zeichenfolgen innerhalb von Verknüpfungen |
| | | LD$> | |
| | | LD$< | |
| | | LD$<> | |
| | | LD$<= | |
| | | LD$>= | |
| | | AND$= | |
| | | AND$> | |
| | | AND$< | |
| | | AND$<> | |
| | | AND$>= | |
| | | AND$<= | |
| | | OR$= | |
| | | OR$> | |
| | | OR$< | |

| Einteilung | | Anweisung | Bedeutung |
|---|---|---|---|
| Vergleichsanweisungen | Vergleich von Zeichenfolgen | OR$<> | Zeichenweiser Vergleich von Zeichenfolgen innerhalb von Verknüpfungen |
| | | OR$<= | |
| | | OR$>= | |
| | Blockweiser Vergleich von Binärdaten | BKCMP= | Verglichen werden die Zeichen, die in aufeinanderfolgenden Operanden (16-Bit-Blöcke) in zwei verschiedenen Datenquellen gespeichert sind. Die Anzahl der 16-Bit-Blöcke wird in der Anweisung festgelegt. Das Vergleichsergebnis wird in einem separaten Bereich abgelegt. |
| | | BKCMP> | |
| | | BKCMP< | |
| | | BKCMP<> | |
| | | BKCMP<= | |
| | | BKCMP>= | |
| Arithmetikanweisungen | Addition und Subtraktion | + | Addition von 16-Bit-Binärdaten |
| | | - | Subtraktion von 16-Bit-Binärdaten |
| | | D+ | Addition von 32-Bit-Binärdaten |
| | | D- | Subtraktion von 32-Bit-Binärdaten |
| | | B+ | Addition von 4-stelligen BCD-Daten |
| | | B- | Subtraktion von 4-stelligen BCD-Daten |
| | | DB+ | Addition von 8-stelligen BCD-Daten |
| | | DB- | Subtraktion von 8-stelligen BCD-Daten |
| | | E+ | Addition von Gleitkommazahlen |
| | | E- | Subtraktion von Gleitkommazahlen |
| | | BK+ | Blockweise Addition von Binärdaten |
| | | BK- | Blockweise Subtraktion von Binärdaten |
| | Multiplikation und Division | x | Multiplikation von 16-Bit-Binärdaten |
| | | / | Division von 16-Bit-Binärdaten |
| | | Dx | Multiplikation von 32-Bit-Binärdaten |
| | | D/ | Division von 32-Bit-Binärdaten |
| | | Bx | Multiplikation von 4-stelligen BCD-Daten |
| | | B/ | Division von 4-stelligen BCD-Daten |
| | | DBx | Multiplikation von 8-stelligen BCD-Daten |
| | | DB/ | Division von 8-stelligen BCD-Daten |
| | | Ex | Multiplikation von Gleitkommazahlen |
| | | E/ | Division von Gleitkommazahlen |
| | Verknüpfung von Zeichenfolgen | S+ | Anfügen einer Zeichenfolge wird an eine andere Zeichenfolge. |
| | Inkrementieren und Dekrementieren von Binärdaten | INC | Inkrementieren (Erhöhung des aktuellen Wert um „1“) von 16-Bit-Binärdaten |
| | | DINC | Inkrementieren von 32-Bit-Binärdaten |
| | | DEC | Dekrementieren (Verminderung des aktuellen Wert um „1“) von 16-Bit-Binärdaten |
| | | DDEC | Dekrementieren von 32-Bit-Binärdaten |
| Konvertierungsanweisungen | Binärdaten -> BCD | BCD | Wandlung von 16-Bit-Binärdaten in BCD-Daten |
| | | DBCD | Wandlung von 32-Bit-Binärdaten in BCD-Daten |
| | | BKBCD | Blockweise Umwandlung von BIN-Daten in BCD-Daten |
| | BCD->Binärdaten | BIN | Wandlung von 4-stelligen BCD-Daten in Binärdaten |
| | | DBIN | Wandlung von 8-stelligen BCD-Daten in Binärdaten |
| | | BKBIN | Blockweise Umwandlung von BCD- in BIN-Daten |
| | Binärzahl -> Gleitkommazahl | FLT | Wandlung einer 16-Bit-Binärzahl in eine Gleitkommazahl |
| | | DFLT | Wandlung einer 32-Bit-Binärzahl in eine Gleitkommazahl |
| | Gleitkommazahl -> Binärzahl | INT | Wandlung einer Gleitkommazahl in eine 16-Bit-Binärzahl |
| | | DINT | Wandlung einer Gleitkommazahl in eine 32-Bit-Binärzahl |
| | Binärdaten -> Binärdaten | DBL | Konvertierung von 16-Bit-Binärdaten in 32-Bit-Binärdaten |
| | | WORD | Konvertierung von 32-Bit-Binärdaten in 16-Bit-Binärdaten |

| Einteilung | | Anweisung | Bedeutung |
|---|---|---|---|
| Konvertierungs-anweisungen | Binärdaten -> Gray-Code | **GRY** | Konvertierung von 16-Bit-Binärdaten in den Gray-Code |
| | | **DGRY** | Konvertierung von 32-Bit-Binärdaten in den Gray-Code |
| | Gray-Code -> Binärdaten | **GBIN** | Konvertierung von Gray-Code-Daten in 16-Bit-Binärdaten |
| | | **DGBIN** | Konvertierung von Gray-Code-Daten in 32-Bit-Binärdaten |
| | Vorzeichenumkehr | **NEG** | Zweierkomplementbildung (Vorzeichenumkehr) von 16-Bit-Binärdaten |
| | | **DNEG** | Zweierkomplementbildung von 32-Bit-Binärdaten |
| | | **ENEG** | Vorzeichenumkehr bei Gleitkommazahlen |
| Transfer-anweisungen | für 16-Bit-Daten | **MOV** | Übertragung von einzelnen 16-Bit-Daten |
| | | **BMOV** | Blockweise Datenübertragung (16-Bit) |
| | | **FMOV** | Füllen eines Datenblocks (Identischer Inhalt in allen Operanden des Datenblocks) |
| | | **XCH** | Inhalte von zwei Operanden tauschen |
| | | **BXCH** | Blockweiser Austausch von Binärdatenblöcken |
| | | **SWAP** | Austausch der Bytes innerhalb eines Wortes |
| | | **EROMWR** | Schreiben von Daten in ein EEPROM-Register |
| | für 32-Bit-Daten | **DMOV** | Übertragung von einzelnen 32-Bit-Daten |
| | | **DXCH** | Inhalte von zwei Operanden tauschen |
| | für Gleitkommazahlen | **EMOV** | Übertragung von Gleitkommazahlen |
| | für Zeichenfolgen | **$MOV** | Übertragung von Zeichenfolgen |
| | Inversion | **CML** | Dateninversion (bitweise Negation) von 16-Bit-Binärdaten |
| | | **DCML** | Dateninversion (bitweise Negation) von 32-Bit-Binärdaten |
| | für Dateien | **SP.FWRITE** | Daten in eine Datei schreiben |
| | | **SP.FREAD** | Daten aus einer Datei lesen |
| | für Datenblöcke | **RBMOV** | Übertragung von Datenblöcken mit hoher Geschwin-digkeit |
| Sprunganweisungen | | **CJ** | Bedingter Sprung innerhalb eines Programms |
| | | **SCJ** | Bedingter Sprung im nächsten Zyklus |
| | | **JMP** | Sprunganweisung |
| | | **GOEND** | Sprung zum Programmende |
| Steuerung von Interrupts | Interrupts freigeben | **EI** | Ermöglicht den Aufruf eines Interrupt-Programms |
| | Interrupts sperren | **DI** | Verhindert die Abarbeitung eines Interrupt-Programms |
| | Einzelne Interrupts freigeben/sperrem | **IMASK** | Steuerung der Ausführungsbedingung von Interrupt-Pro-grammen |
| | Ende des Interrupt-Programms | **IRET** | Rücksprung aus dem Interrupt-Programm ins Hauptpro-gramm |
| Aktualisierung von Daten | Ein- und Ausgänge | **RFS** | Aktualisierung der Ein- und Ausgänge eines bestimmten Bereichs für einen Programmzyklus |
| | Netzwerk- und Schnittstellendaten | **COM** | Aktualisierung der Netzwerk- und Schnittstellendaten |
| | Link-Refresh | **DI** | Verhindern der Link-Refresh-Ausführung |
| | | **EI** | Ermöglichen der Link-Refresh-Ausführung |
| Logik-anweisungen | UND-Logik | **WAND** | Verknüpfung von zwei 16-Bit-Operanden |
| | | **DAND** | Verknüpfung von zwei 32-Bit-Operanden |
| | | **BKAND** | Verknüpfung von 16-Bit-Operanden in Datenblöcken |
| | ODER-Logik | **WOR** | Verknüpfung von zwei 16-Bit-Operanden |
| | | **DOR** | Verknüpfung von zwei 32-Bit-Operanden |
| | | **BKOR** | Verknüpfung von 16-Bit-Operanden in Datenblöcken |
| | Exklusiv-ODER-Logik | **WXOR** | Verknüpfung von zwei 16-Bit-Operanden |
| | | **DXOR** | Verknüpfung von zwei 32-Bit-Operanden |
| | | **BKXOR** | Verknüpfung von 16-Bit-Operanden in Datenblöcken |

| Einteilung | | Anweisung | Bedeutung |
|---|---|---|---|
| Logik-anweisungen | Exklusiv-NOR-Logik | WNXR | Verknüpfung von zwei 16-Bit-Operanden |
| | | DNXR | Verknüpfung von zwei 32-Bit-Operanden |
| | | BKXNR | Verknüpfung von 16-Bit-Operanden in Datenblöcken |
| Rotations-anweisungen | 16-Bit-Daten | ROR | Rotation von Bits nach rechts |
| | | RCR | Rotieren von Bits nach rechts mit Carry-Bit |
| | | ROL | Rotation von Bits nach links |
| | | RCL | Rotieren von Bits nach links mit Carry-Bit |
| | 32-Bit-Daten | DROR | Rotation von Bits nach rechts |
| | | DRCR | Rotieren von Bits nach rechts mit Carry-Bit |
| | | DROL | Rotation von Bits nach links |
| | | DRCL | Rotieren von Bits nach links mit Carry-Bit |
| Verschiebe-anweisungen | 16-Bit-Daten | SFR | Verschiebung um n Bit nach rechts (n: 0 bis 15) |
| | | SFL | Verschiebung um n Bit nach links (n: 0 bis 15) |
| | Bit-Operanden | BSFR | Schieben einer Anzahl Bit-Operanden um 1 Bit nach rechts |
| | | BSFL | Schieben einer Anzahl Bit-Operanden um 1 Bit nach links |
| | Wort-Operanden | DSFR | Schieben einer Anzahl Wort-Operanden um 1 Adresse nach rechts bzw. links |
| | | DSFL | |
| Bit-Verarbei-tungsanweisun-gen | Setzen/Rücksetzen | BSET | Setzen einzelner Bits |
| | | BRST | Zurücksetzen einzelner Bits |
| | | BKRST | Zurücksetzen von Bitbereichen |
| | Zustandsabfrage | TEST | Zustandsabfrage einzelner Bits in 16-/32-Bit-Daten-wörtern |
| | | DTEST | |
| Datenverarbei-tungsanweisun-gen | Suchen | SER | Suchen von 16-Bit-Daten |
| | | DSER | Suchen von 32-Bit-Daten |
| | Prüfen | SUM | Bestimmung der Anzahl der gesetzten Bits in einem 6-/32-Bit-Datenwort |
| | | DSUM | |
| | Decodieren | DECO | Decodierung von 8 nach 256 Bit (binär in dezimal) |
| | Codieren | ENCO | Codierung von 256 nach 8 Bit (dezimal in binär) |
| | 7-Segment-Codierung | SEG | Wandlung eines 4-stelligen Binärcodes zur Ansteuerung einer 7-Segment-Anzeige |
| | 16-Bit-Datenworte auftrennen oder zusammenführen | DIS | Auftrennen von 16-Bit-Datenwerten in Gruppen zu 4 Bits |
| | | UNI | Speicherung der jeweils 4 niedrigstwertigen Bits von bis zu vier 16-Bit-Datenwerten in einem 16-Bit-Datenwert |
| | | NDIS | Trennen von Daten in Bit-Gruppen variabler Größe |
| | | NUNI | Gruppieren von Daten in Bit-Gruppen variabler Größe |
| | | WTOB | Trennen von Daten in Byte-Gruppen |
| | | BTOW | Gruppieren von Daten in Byte-Gruppen |
| | Suchen von Maximal-werten | MAX | Suche des größten Wertes in 16-Bit-Datenblöcken |
| | | DMAX | Suche des größten Wertes in 32-Bit-Datenblöcken |
| | Suchen von Minimal-werten | MIN | Suche des kleinsten Wertes in 16-Bit-Datenblöcken |
| | | DMIN | Suche des kleinsten Wertes in 32-Bit-Datenblöcken |
| | Sortieren | SORT | Sortieren von 16-Bit-Daten |
| | | DSORT | Sortieren von 32-Bit-Daten |
| | Summenbildung | WSUM | Summenbildung von 16-Bit-Binärdaten |
| | | DWSUM | Summenbildung von 32-Bit-Binärdaten |
| Strukturierte Programman-weisungen | Wiederholungen | FOR | Anfang einer Programmwiederholung |
| | | NEXT | Ende einer Programmwiederholung |
| | | BREAK | Beenden der FOR-NEXT-Ausführung |

| Einteilung | | Anweisung | Bedeutung |
|---|---|---|---|
| Strukturierte Programmanweisungen | Unterprogramme | **CALL** | Aufruf eines Unterprogramms |
| | | **RET** | Ende eines Unterprogramms |
| | | **FCALL*** | Ausgänge in Unterprogrammen zurücksetzen |
| | | **ECALL*** | Aufruf eines Unterprogramms, das sich in einem anderen Programm befindet |
| | | **EFCALL*** | Ausgänge in Unterprogrammen zurücksetzen, die sich in einem anderen Programm befinden |
| | Index-Bearbeitung | **IX** | Indizierte Adressierung eines Programmteils |
| | | **IXEND** | |
| | | **IXDEV** | Speicherung indizierter Operandenadressen in einer Index-Liste |
| | | **IXSET** | |
| Verarbeitungsanweisungen für Datenlisten | Schreiben | **FIFW** | Daten in eine Datenliste schreiben |
| | Lesen | **FIFR** | Lesen zuerst eingegebener Daten aus einer Datenliste |
| | | **FPOP** | Lesen zuletzt eingegebener Daten aus einer Datenliste |
| | Löschen | **FDEL** | Löschen bestimmter Datenblöcke in einer Datenliste |
| | Einfügen | **FINS** | Einfügen bestimmter Datenblöcke in die Datenliste |
| Anweisungen für den Pufferspeicherzugriff | Lesen | **FROM** | 16-Bit-Daten aus Sondermodul lesen |
| | | **DFRO** | 32-Bit-Daten aus Sondermodul lesen |
| | Schreiben | **TO** | 16-Bit-Daten in Sondermodul schreiben |
| | | **DTO** | 32-Bit-Daten in Sondermodul schreiben |
| Display-Anweisungen | ASCII-Ausgabe | **PR** | Ausgabe einer ASCII-Zeichenfolge an ein peripheres Gerät |
| | | **PRC** | Ausgabe eines Kommentars im ASCII-Code an ein peripheres Gerät |
| | Anzeige löschen | **LEDR** | Rücksetzen von Fehlermerkern und der LED-Anzeige |
| Fehlererkennung und -beseitigung | Fehlerkontrolle | **CHKST** | Startanweisung zur CHK-Anweisung |
| | | **CHK** | Fehlerkontrolle |
| | | **CHKCIR** | Prüfnetzwerke für die CHK-Anweisung erzeugen |
| | | **CHKEND** | End-Anweisung für den Programmbereich mit den erzeugten Prüfnetzwerken |
| | Speicherung von Operandenzuständen | **SLT** | Status Latch setzen (Operandenzustände speichern) |
| | | **SLTR** | Status Latch rücksetzen (Operandenzustände löschen) |
| | Abtastüberwachung (Sampling Trace) | **STRA** | Abtastüberwachung setzen |
| | | **STRAR** | Abtastüberwachung zurücksetzen |
| | Programmüberwachung (Program Trace) | **PTRA** | Programmüberwachung setzen |
| | | **PTRAR** | Programmüberwachung zurücksetzen |
| | | **PTRAEXE** | Programmüberwachung ausführen |
| | Überwachung (Trace) | **TRACE** | Überwachung starten |
| | | **TRACER** | Die durch die TRACE-Anweisung gespeicherten Daten werden gelöscht. |
| Verarbeitungsanweisungen für Zeichenfolgen | Binär -> Dezimal (ASCII) | **BINDA** | Konvertierung von 16-/32-Bit-Binärdaten in Dezimalzahlen im ASCII-Code |
| | | **DBINDA** | |
| | Binär -> Hexadezimal (ASCII) | **BINHA** | Konvertierung von 16-/32-Bit-Binärdaten in Hexadezimalzahlen im ASCII-Code |
| | | **DBINHA** | |
| | BCD -> ASCII | **BCDDA** | Wandlung von 4-stelligen BCD-Daten in den ASCII-Code |
| | | **DBCDDA** | Wandlung von 8-stelligen BCD-Daten in den ASCII-Code |
| | Dezimal (ASCII) -> Binär | **DABIN** | Wandlung dezimaler ASCII-Daten in 16-/32-Bit-Binärdaten |
| | | **DDABIN** | |
| | Hexadezimal (ASCII) -> Binär | **HABIN** | Wandlung hexadezimaler ASCII-Daten in 16-/32-Bit-Binärdaten |
| | | **DHABIN** | |

* Die Anweisungen FCALL, ECALL und EFCALL können nicht mit der Programmier-Software GX IEC Developer programmiert werden.

| Einteilung | | Anweisung | Bedeutung |
|---|---|---|---|
| Verarbeitungs-anweisungen für Zeichen-folgen | Dezimal (ASCII) -> BCD | **DABCD** | Konvertierung dezimaler ASCII-Daten in 4-stellige BCD-Daten |
| | | **DDABCD** | Konvertierung dezimaler ASCII-Daten in 8-stellige BCD-Daten |
| | Auslesen von Kom-mentardaten | **COMRD** | Kommentar lesen und als ASCII-Code speichern |
| | Längenerfassung | **LEN** | Länge von Zeichenfolgen erfassen |
| | Binär -> Zeichenfolge | **STR** | Einfügen eines Dezimalkommas und Konvertierung von 16-/32-Bit-Binärdaten in Zeichenfolgen |
| | | **DSTR** | |
| | Zeichenfolge -> Binär | **VAL** | Konvertierung von Zeichenfolgen in 16-/32-Bit-Binärdaten |
| | | **DVAL** | |
| | Gleitkommazahlen -> Zeichenfolge | **ESTR** | Konvertierung von Gleitkommazahlen in eine Zeichen-folge |
| | Zeichenfolge -> Gleit-kommazahl | **EVAL** | Konvertierung einer Zeichenfolge in eine dezimale Gleit-kommazahlen |
| | Gleitkommazahlen -> BCD | **EMOD** | Gleitkommazahlumrechnung in das BCD-Format |
| | Gleitkommazahlen -> Dezimal | **EREXP** | BCD-Gleitkommazahlumrechnung in das Dezimal-Format |
| | BIN-16-Bit-Daten -> ASCII | **ASC** | Konvertierung von BIN-16-Bit-Daten in den ASCII-Code |
| | ASCII -> Binär | **HEX** | Wandlung von hexadezimalen ASCII-Werten in Binärwerte |
| | Auszug von Zeichen-folgendaten | **RIGHT** | Auszug der Daten vom rechten Teil der Zeichenfolge |
| | | **LEFT** | Auszug der Daten vom linken Teil der Zeichenfolge |
| | Speichern | **MIDR** | Speichern definierter Zeichenfolgeteile |
| | Verschieben | **MIDW** | Verschieben von Zeichenfolgeteilen in einen definierten Bereich |
| | Suche | **INSTR** | Suche von Zeichenfolgen |
| Anweisungen für Gleitkom-mazahlen | Trigonometrische Funktionen | **SIN** | Sinusberechnung |
| | | **COS** | Cosinusberechnung |
| | | **TAN** | Tangensberechnung |
| | | **ASIN** | Arcussinusberechnung |
| | | **ACOS** | Arcuscosinusberechnung |
| | | **ATAN** | Arcustangensberechnung |
| | | **RAD** | Umrechnung von Grad in Radiant |
| | | **DEG** | Umrechnung von Radiant in Grad |
| | Arithmetische Funktionen | **SQR** | Quadratwurzelberechnung |
| | | **EXP** | Gleitkommazahl als Exponent zur Basis e |
| | | **LOGE** | Logarithmus-naturalis-Berechnung |
| Sonder-funktionen | Zufallszahlen | **RND** | Generierung von Zufallszahlen |
| | | **SRND** | Aktualisierung von Zufallszahlenserien |
| Anweisungen für BCD-Daten | Trigonometrische Funktionen | **BSIN** | Sinusberechnung |
| | | **BCOS** | Cosinusberechnung |
| | | **BTAN** | Tangensberechnung |
| | | **BASIN** | Arcussinusberechnung |
| | | **BACOS** | Arcuscosinusberechnung |
| | | **BATAN** | Arcustangensberechnung |
| | Arithmetische Funktionen | **BSQR** | Quadratwurzelberechnung aus 4-stelligen BCD-Daten |
| | | **BDSQR** | Quadratwurzelberechnung aus 8-stelligen BCD-Daten |

| Einteilung | | Anweisung | Bedeutung |
|---|---|---|---|
| Datenkontroll-anweisungen | Begrenzung | LIMIT | Begrenzung des Ausgangswertebereichs von 16-/32-Bit-Binärdaten |
| | | DLIMIT | |
| | Eingangs-Offset | BAND | Eingangsoffset-Wert von 16-/32-Bit-Binärdaten festlegen |
| | | DBAND | |
| | Ausgangs-Offset | ZONE | Ausgangsoffset-Wert von 16-/32-Bit-Binärdaten festlegen |
| | | DZONE | |
| Anweisungen für File-Register | Umschaltan-weisungen | RSET | Umschaltung zwischen File-Registerblöcken |
| | | QDRET | Umschaltung zwischen Dateien in File-Registern |
| | | QCDSET | Umschaltung zwischen Dateien für Kommentardaten in File-Registern |
| | Lesen | ZRRDB | Direktes Lesen eines Bytes aus einem File-Register |
| | Schreiben | ZRWRB | Direktes Schreiben eines Bytes in einem File-Register |
| Operationen mit der inte-grierten Uhr der SPS | Lesen | DATERD | Uhrzeit und Datum lesen |
| | Stellen | DATEWR | Uhrzeit und Datum in die SPS übertragen |
| | Addieren | DATE+ | Addition von Uhrdaten |
| | Subtraieren | DATE- | Subtraktion von Uhrdaten |
| | Formatumwandlung | SECOND | Zeitangabe in der Form „Stunden, Minuten, Sekunden“ in Sekunden wandeln |
| | | HOUR | Zeitangabe in Sekunden in das Format „Stunden, Minu-ten, Sekunden“ wandeln |
| Anweisungen für Peripherie-geräte | Ausgabe | MSG | Ausgabe von Meldungen an Peripheriegeräte |
| | Eingabe | PKEY | Tastatureingabe von Daten an Peripheriegeräten |
| | | KEY | Tastatureingabe numerischer Werte |
| Anweisungen zur Steuerung der Programm-ausführung | Standby-Modus | PSTOP | Umschaltung eines Programms in den Standby-Modus |
| | | POFF | Umschaltung eines Programms in den Standby-Modus mit Rücksetzen der Ausgänge |
| | Zyklische Programm-ausführung | PSCAN | Umschaltung eines Programms in den Modus einer Pro-grammausführung pro Zyklus |
| | Niedrige Verarbei-tungsgeschwindigkeit | PLOW | Umschaltung eines Programms in den Modus niedriger Verarbeitungsgeschwindigkeit |
| Anweisungen zur Handha-bung von Pro-grammen | Programm laden | PLOADP | Programm aus Speicher laden |
| | Programm löschen | PUNLOADP | Programm, das im Standby-Modus ist, löschen |
| | Löschen und Laden | PSWAPP | Programm, das im Standby-Modus ist, löschen und Pro-gramm aus Speicher laden |
| Anweisungen für den Daten-austausch in Netzwerken | Datenaktualisierung | ZCOM | Datenaktualisierung in Netzwerkmodulen |
| | Routing | RTREAD | Lesen der Routing-Informationen des Netzwerks |
| | | RTWRITE | Schreiben der Routing-Informationen des Netzwerks |
| Anweisungen zum Datenaus-tausch im Multi-CPU-Betrieb | Daten schreiben | S.TO | Daten in den gemeinsamen Speicherbereich eintragen |
| | Daten lesen | FROM | Daten aus dem gemeinsamen Speicherbereich einer anderen CPU lesen |
| | Daten aktualisieren | COM | Der gemeinsame Speicherbereich für den Multi-CPU-Betrieb wird aktualisiert. |
| System-steuerung | Watch-Dog-Timer | WDT | Watch-Dog-Timer zurücksetzen |
| | Modulinformationen | UNIRD | Informationen aus einem Modul lesen |
| | Indexregister | ZPUSH | Sichern der Indexregisterinhalte in ein Register |
| | | ZPOP | Wiederherstellen der Indexregisterinhalte aus einem Register |
| | Operandenadresse speichern | ADRSET | Speichern einer indirekten Adresse (nicht bei GX IEC Developer) |
| | Systemtakte | DUTY | Vorgabe von Ausführungszyklen eines Operanden |

| Einteilung | | Anweisung | Bedeutung |
|---|---|---|---|
| Anwendungs-bezogene Anweisungen | Zähler | **UDCNT1** | Einphasiger Auf-/Abwärtszähler |
| | | **UDCNT2** | Zweiphasiger Auf-/Abwärtszähler |
| | Timer | **TTMR** | Programmierbarer Timer |
| | | **STMR** | Sonderfunktions-Timer (langsame Timer) |
| | | **STMRH** | Sonderfunktions-Timer (schnelle Timer) |
| | Rotations-tischanweisung | **ROTC** | Positionieranweisung für Rotationstische |
| | Rampensignal | **RAMP** | Stufenweise Erhöhung eines Wertes |
| | Impulszähler | **SPD** | Zählung von Eingangsimpulse für eine angegebene Dauer und Speicherung des Zählwerts. |
| | Impulsausgabe | **PLSY** | Impulsausgang mit einstellbarer Anzahl von Impulsen |
| | Pulsweitenmodulation | **PWM** | Ausgabe von Impulsen; Periodendauer und Impulslänge sind einstellbar |
| | Eingabematrix | **MTR** | Bildung einer Matrix zum Einlesen von Informationen |
| Anweisungen für serielle Schnittstellen-module | Daten lesen | **BUFRCVS** | Daten vom Schnittstellenmodul in die SPS-CPU übertragen |
| | Daten senden | **PRR** | Daten über das Schnittstellenmodul mittels anwenderdefinierter Datenrahmen senden |
| | Anwenderdefinierte Datenrahmen | **GETE** | Anwenderdefinierte Datenrahmen lesen |
| | | **PUTE** | Anwenderdefinierte Datenrahmen schreiben oder löschen |
| Anweisungen für PROFI-BUS/DP-Module | Daten lesen | **BBLKRD** | Daten aus dem Pufferspeicher eines PROFIBUS/DP-Moduls lesen und in der SPS-CPU speichern |
| | Daten schreiben | **BBLKWR** | Daten aus der SPS-CPU in den Pufferspeicher eines PROFIBUS/DP-Moduls eintragen |
| Anweisungen für ETHERNET-Module | Daten lesen | **BUFRCV** | Die bei der Kommunikation mit festen Puffern empfangenen Daten werden aus dem ETHERNET-Modul gelesen. |
| | | **BUFRCVS** | |
| | Daten schreiben | **BUFSND** | Daten aus der CPU zum ETHERNET-Modul übertragen |
| | Verbindung öffnen | **OPEN** | Aufbau einer Verbindung |
| | Verbindung schließen | **CLOSE** | Abbau einer Verbindung |
| | Fehler löschen | **ERRCLR** | Fehlercodes in Pufferspeicher löschen, LED „ERR." des ETHERNET-Moduls ausschalten |
| | Fehlercode lesen | **ERRRD** | Fehlercodes aus Pufferspeicher lesen |
| | Initialisierung | **UINI** | Erneute Initialisierung des ETHERNET-Moduls |
| Anweisung für CC-Link | Netzwerkparameter übertragen | **RLPASET** | Übertragung der Netzwerkparameter in die Master-Station des CC-Link |
| | Daten lesen | **RIRD** | Daten aus dem Pufferspeicher eines CC-Link-Moduls einer anderen Station oder aus der SPS-CPU dieser Station lesen |
| | | **RICV** | Daten aus dem Pufferspeicher einer intelligenten CC-Link-Station unter Verwendung eines Handshakes lesen |
| | | **RIFR** | Daten lesen, die von einer anderen Station in den automatisch aktualisierten Bereich des Pufferspeichers der CC-Link-Master-Station eingetragen wurden |
| | Daten schreiben | **RIWT** | Daten in den Pufferspeicher eines CC-Link-Moduls einer anderen Station oder in die SPS-CPU dieser Station schreiben |
| | | **RISEND** | Daten in den Pufferspeicher einer intelligenten CC-Link-Station unter Verwendung eines Handshakes eintragen |
| | | **RITO** | Daten aus der SPS-CPU in den automatisch aktualisierten Bereich des Pufferspeichers der CC-Link-Master-Station eintragen. Anschließend werden diese Daten zu der angegebenen Station übertragen. |

### 6.1.1 Zusätzliche Anweisungen für Prozess-CPUs

Zur schnellen und einfachen Programmierung von Regelungen können bei den Prozess-CPUs Q12PHCPU und Q25PHCPU die in der folgenden Tabelle aufgeführten Regelungsanweisungen eingesetzt werden.

| Einteilung | | Anweisung | Bedeutung |
|---|---|---|---|
| Ein- und Ausgabeanweisungen | Eingabe | **IN** | Verarbeitung eines Eingangswertes (Istwert) |
| | Ausgabe | **OUT1** | Ausgabe der Stellgröße |
| | | **OUT2** | |
| | Manuelle Ausgabe | **MOUT** | Ausgabe der Stellgröße im Handbetrieb der Regelung |
| | PWM | **DUTY** | Pulsweitenmoduliertes Signal (0 bis 100 %) ausgeben |
| | Vergleich | **BC** | Vergleich eines Eingangswertes mit bis zu zwei Sollwerten und Ausgabe des Ergebnisses durch Bit-Operanden |
| | Impulsintegrierung | **PSUM** | Integration eines Eingangssignals, Bereichsüberwachung und Ausgabe des Ergebnisses |
| Regelung | PID-Regler | **PID** | PID-Regelung |
| | | **2PID** | PID-Regelung mit zwei zusätzlichen Einstellmöglichkeiten |
| | | **PIDP** | PID-Regelung mit Grenzwertprüfung |
| | PI-Regler | **SPI** | PI-Regelung mit Abtastung |
| | I-PD-Regler | **IPD** | I-PD-Regelung |
| | PI-Regler | **BPI** | PI-Regelung |
| | Zweipunkt-Regler | **ONF2** | Zweipunkt-Regelung |
| | Dreipunkt-Regler | **ONF3** | Dreipunkt-Regelung |
| Signalaufbereitung | Anstiegsbegrenzer | **R** | Begrenzung der Anstiegsgeschwindigkeit des Ausgangssignals |
| | Grenzwertalarme | **PHPL** | Prüfung eines Eingangswertes und Ausgabe von Alarmen bei Überschreitung eines unteren/oberen Grenzwerts. |
| | Voreilung/Verzögerung | **LLAG** | Der Ausgang der LLAG-Anweisung folgt entweder dem Eingang verzögert oder eilt dem Eingang vor. |
| | Integrierer | **I** | Integration eines Eingangssignals und Ausgabe des Ergebnisses |
| | Differenzierer | **D** | Differenzierung eines Eingangssignals und Ausgabe des Ergebnisses |
| | Einstellbare Totzeit | **DED** | Ausgabe eines Eingangswertes nach Ablauf von Totzeiten |
| | Höchsten/Mittleren/Niedrigsten Wert ausgeben | **HS** | Höchsten von bis zu 16 Eingangswerten ausgeben |
| | | **LS** | Niedrigsten von bis zu 16 Eingangswerten ausgeben |
| | | **MID** | Aus bis zu 16 Eingangswerten wird der Mittlere ausgewählt und ausgegeben. |
| | Mittelwertbildung | **AVE** | Bildung des arithmetischen Mittelwertes aus max. 16 Eingangswerten |
| | Wertbegrenzung | **LIMT** | Begrenzung eines Eingangswertes auf einen Bereich, der durch zwei Grenzwerte gebildet wird |
| | Rampenbildung | **VLMT1** | Begrenzung der Änderungsgeschwindigkeit des Ausgangswertes |
| | | **VLMT2** | |
| | Einstellbare Totzone | **DBND** | Ein Eingangswert, der sich innerhalb der eingestellten Totzone befindet, wird nicht als Ausgangswert ausgegeben. |
| | Programmierbare Sollwertausgabe | **PGS** | Ausgangswerte nach einem vorgegebenen Muster ausgeben |
| | Umschaltung zwischen zwei Eingangssignalen | **SEL** | Umschaltung zwischen zwei Eingangssignalen, deren Signale im Automatikbetrieb ausgegeben werden. Im Handbetrieb wird die Stellgröße aus dem Regelungskennsatz ausgegeben. |
| | Stoßfreie Umschaltung | **BUMP** | Bei der Umschaltung vom Hand- in den Automatikbetrieb werden die Sollwerte angepasst. |
| | Analoger Speicher | **AMR** | Der Ausgangswert wird in konstanten Schritten verändert. |

| Einteilung | | Anweisung | Bedeutung |
|---|---|---|---|
| Anweisungen zur Konvertierung und Kompensation | Werte skalieren | FG | Der Ausgangswert hängt vom Eingangswert und einem von Anwender eingegebenen Kurvenverlauf ab. |
| | | IFG | |
| | Filter | **FLT** | Erfassung eines Eingangswertes in einstellbaren Intervallen und Mittelwertbildung |
| | Summierer | **SUM** | Summierung eines Eingangswertes und Ausgabe des Ergebnisses |
| | Temperatur-/ Druckkompensation | **TPC** | Eingangswert mit einem Temperatur- und/oder einem Druckkorrekturwert verrechnen und das Ergebnis ausgeben |
| | Wandlung in normierten Wert | **ENG** | Wandlung eines Eingangswertes in der Einheit % in einen normierten Wert mit einer pysikalischen Einheit |
| | Rückwandlung eines normierten Wertes | **IENG** | Wandlung eines Eingangswertes mit einer pysikalischen Einheit in einen prozentualen Wert |
| Arithmetische Anweisungen | Addition | **ADD** | Rechenoperationen, bei denen zusätzliche Koeffizienten angegeben werden können |
| | Subtraktion | **SUB** | |
| | Multiplikation | **MUL** | |
| | Division | **DIV** | |
| | Extraktion (Wurzelbildung) | **SQR** | Berechnung der Quadratwurzel des Eingangswertes |
| | Absoluten Wert ausgeben | **ABS** | Betrages des Eingangswertes bilden und ausgeben |
| Vergleichsanweisungen | Vergleich auf „größer als“ | **> (GT)** | Vergleich zweier Eingangswerte unter Berücksichtigung einer Hysterese |
| | Vergleich auf „kleiner als“ | **< (LT)** | |
| | Vergleich auf „gleich“ | **= (EQ)** | |
| | Vergleich auf „größer oder gleich“ | **>= (GE)** | |
| | Vergleich auf „kleiner oder gleich“ | **<= (LE)** | |
| Autom. Ermittlung der Reglerparameter | Autotuning | **AT1** | Automatische Ermittlung der Parameter für eine Regelung, die mit der PID-oder 2PID-Anweisung realisiert ist |

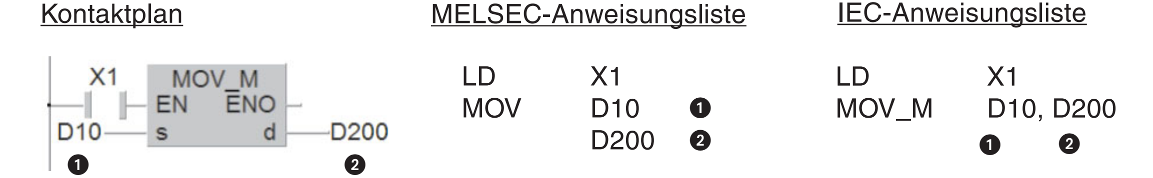
**HINWEIS** Eine ausführliche Beschreibung der Regelungsanweisungen enthält die Programmieranleitung zur QnPHCPU, Art.-Nr. 158626.

# 6.2 Anweisungen für den Transfer von Daten

In der SPS dienen Datenregister als Speicher für Mess- und Ausgabewerte, Zwischenergebnisse oder Tabellenwerte. Zwar lesen zum Beispiel die arithmetischen Anweisungen ihre Operandenwerte direkt aus Datenregistern und tragen – wenn gewünscht – auch das Ergebnis dort ein, zur Unterstützung dieser Anweisungen werden aber Transferanweisungen benötigt, mit denen Daten von einem Register in ein anderes kopiert oder Konstanten in Datenregister eingetragen werden können.

## 6.2.1 Transfer einzelner Daten mit einer MOV-Anweisung

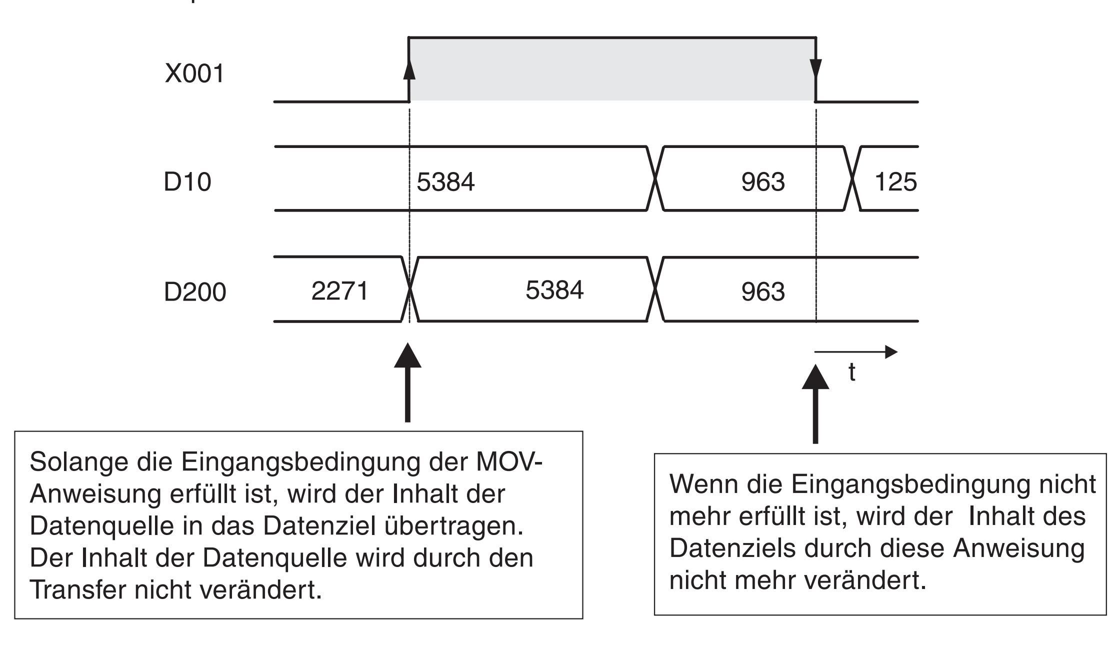
Mit einer MOV-Anweisung (von engl. *to move* = bewegen) werden Daten „bewegt“ und aus einer Datenquelle in ein Ziel kopiert. Der Inhalt der Datenquelle verändert sich dabei nicht.




❶ Datenquelle (Hier kann auch eine Konstante angegeben werden.) Das „s“ in Kontaktplan-Anweisungen steht für *source* = Quelle.

❷ Datenziel; In Kontaktplan-Anweisungen bedeutet „d“ *destination* = Ziel.

In diesem Beispiel wird der Inhalt des Datenregisters D10 in das Datenregister D200 übertragen, wenn der Eingang X1 eingeschaltet ist. Die folgende Abbildung zeigt den Signalverlauf für dieses Beispiel.




### Flankengesteuerte Ausführung der MOV-Anweisung

Für bestimmte Anwendungen ist es vorteilhafter, wenn das Datenziel nur in einem Programmzyklus beschrieben wird. Zum Beispiel, wenn an anderer Stelle im Programm in dasselbe Ziel transferiert wird oder der Transfer nur zu einem definierten Zeitpunkt stattfinden soll.

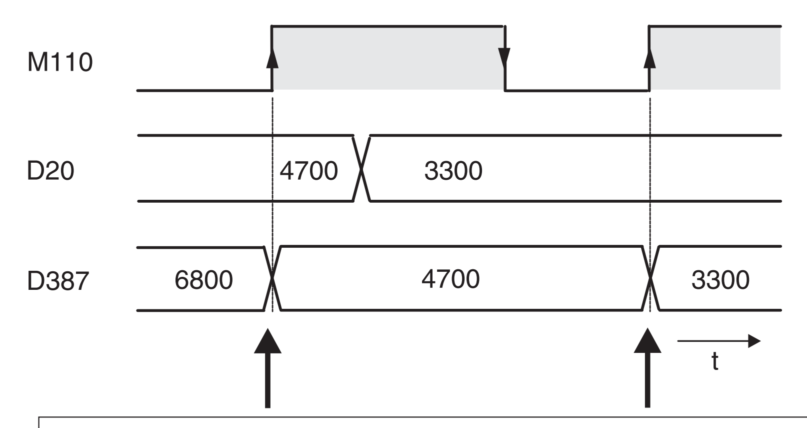
Eine MOV-Anweisung wird bei steigender Flanke der Eingangsbedingung nur **einmal** ausgeführt, wenn der Abkürzung MOV ein „P“ nachgestellt wird. (Der Buchstabe „P“ bezieht sich auf

den englischen Begriff *Pulse* und weist darauf hin, dass die Anweisung durch einen Signalwechsel oder Impuls gesteuert wird.)

Im folgenden Beispiel wird nur dann der Inhalt von D20 in das Datenregister D387 eingetragen, wenn der Signalzustand von M110 von „0“ nach „1“ wechselt.

| Kontaktplan | MELSEC-Anweisungsliste | | IEC-Anweisungsliste | |
|---|---|---|---|---|
| M110 MOVP_M EN ENO D20 s d D387 | LD | M110 | LD | M110 |
| | MOVP | D20 | MOVP_M | D20, D387 |
| | | D387 | | ❶ ❷ |

Auch wenn M110 gesetzt bleibt, wird nicht weiter in das Register D387 transferiert. Der Signalverlauf zu diesen Beispiel macht das deutlich:




Der Inhalt der Datenquelle wird nur bei der steigenden Flanke der Eingangsbedingung in das Datenziel übertragen.

**Transfer von 32-Bit-Daten**

Wenn mit einer MOV-Anweisung 32-Bit-Daten transferiert werden sollen, wird der Anweisung ein „D“ vorangestellt.

Kontaktplan

X1 DMOV_M EN ENO var_D0 s d var_D40

MELSEC-Anweisungsliste

| | | |
|---|---|---|
| LD | X1 | |
| DMOV | D0 | D40 |

IEC-Anweisungsliste

| | | |
|---|---|---|
| LD | X1 | |
| DMOV_M | var_D0, | varD40 |

Wenn der Eingang X1 eingeschaltet ist, wird der Inhalt der Register D0 und D1 in die Datenregister D40 und D41 übertragen (Der Inhalt von D0 wird in D40 kopiert und der Inhalt von D1 nach D41).

**HINWEIS**

Beim GX IEC Developer können 32-Bit-Operanden bei der Programmierung in Kontaktplan und in der IEC-Anweisungsliste nicht direkt angegeben werden. Diese Operanden müssen zuvor als Variablen deklariert werden (siehe Abschnitt 4.6.2). Darauf soll die Bezeichnung var_D0 und var_D40 der Operanden in diesen Beispiel hinweisen.

Auch die Kombination von Doppelwortverabeitung und flankengesteuerter Ausführung ist möglich, wie das folgende Beispiel zeigt.

Kontaktplan

M10
DMOVP_M
EN ENO
var_D10 s d var_D610

MELSEC-Anweisungsliste

| | | |
|---|---|---|
| LD | M10 | |
| DMOVP | D10 | D610 |

IEC-Anweisungsliste

| | |
|---|---|
| LD | X1 |
| DMOVP_M | var_D10, var_D610 |

Beim Setzen des Merkers M10 wird der Inhalt der Register D10 und D11 in die Register D610 und D611 transferiert.

**HINWEIS**

Bitte beachten Sie, dass bei der Programmierung in Kontaktplan und in der IEC-Anweisungsliste 32-Bit-Operanden als Variable deklariert werden müssen (siehe Abschnitt 4.6.2). Diese Operanden können nicht direkt mit der Anweisung eingegeben werden.

## 6.2.2 Transfer von Bit-Operanden in Gruppen

Im vorherigen Abschnitt wurde gezeigt, wie mit einer MOV-Anweisung Konstanten oder die Inhalte von Datenregistern in andere Datenregister übertragen werden können. Aber auch in aufeinanderfolgenden Bit-Operanden, wie Merkern, können numerische Werte gespeichert sein. Um mehrere aufeinanderfolgende Bit-Operanden mit einer Applikationsanweisung anzusprechen, wird die Adresse des ersten Bit-Operanden zusammen mit einem Faktor „K“ angegeben, der die Anzahl der Operanden angibt.

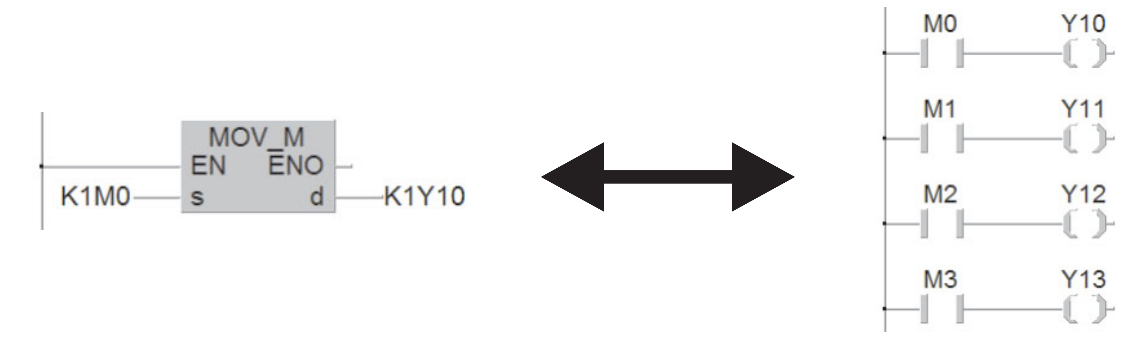
Dieser Faktor „K“ gibt die Anzahl der Einheiten zu je 4 Operanden an: K1 = 4 Operanden, K2 = 8 Operanden, K3 = 12 Operanden usw.

Mit der Angabe „K2M0“ werden zum Beispiel die acht Merker M0 bis M7 definiert. Möglich sind Faktoren von K1 (4 Operanden) bis K8 (32 Operanden).

**Beispiele für die Angabe von Bit-Operanden**

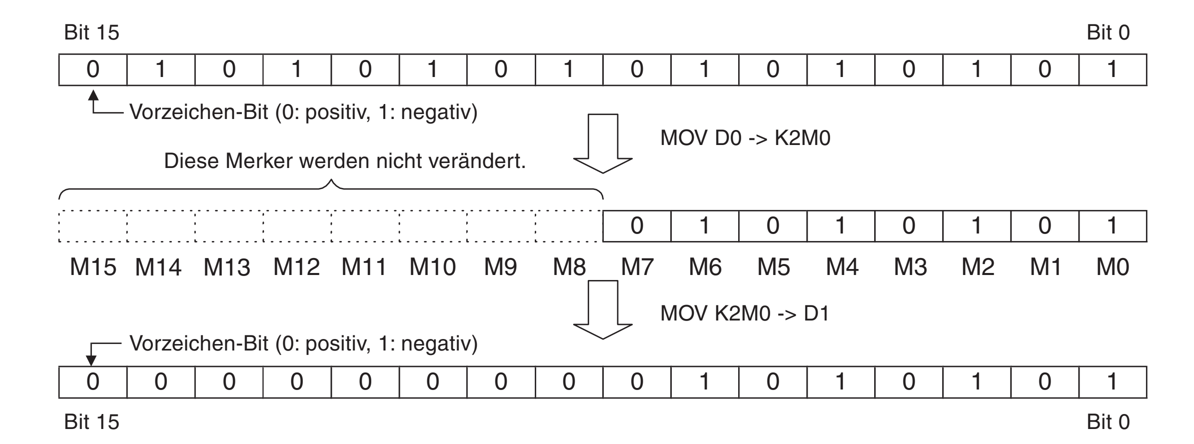
- K1X0: 4 Eingänge, Start bei X0 (X0 bis X3)
- K2X4: 8 Eingänge, Start bei X4 (X4 bis X1B, hexadezimale Zählweise!)
- K4M16: 16 Merker, Start bei M16 (M16 bis M31)
- K3Y0: 12 Ausgänge, Start bei Y0 (Y0 bis Y1B, hexadezimale Zählweise!)
- K8M0: 32 Merker, Start bei M0 (M0 bis M31)

Die Möglichkeit, mehrere Bit-Operanden mit nur einer Anweisung anzusprechen, reduziert auch den Programmieraufwand. Die folgenden beiden Programmsequenzen haben dieselbe Funktion: Das Übertragen der Signalzustände der Merker M0 bis M3 zu den Ausgängen Y10 bis Y13.



Wenn das Datenziel kleiner ist als die Datenquelle, werden die überzähligen Bits nicht übertragen (siehe folgende Abbildung, oberes Beispiel).

Ist das Datenziel größer als die Datenquelle, werden die fehlenden Stellen mit „0“ aufgefüllt. Durch die Interpretation von Bit 15 als Vorzeichen ist der dadurch entstehende Wert immer positiv. (Wie beim unteren Beispiel in der folgenden Abbildung.)



## 6.2.3 Transfer von zusammenhängenden Daten mit einer BMOV-Anweisung

Mit der im Abschnitt 6.2.1 vorgestellten MOV-Anweisung kann maximal ein 16- oder 32-Bit-Wert in ein Datenziel transferiert werden. Für den Transfer von zusammenhängenden Daten könnten mehrere MOV-Anweisungen nacheinander programmiert werden. Um Ihnen diesen Programmieraufwand zu ersparen, steht Ihnen die BMOV-Anweisung zur Verfügung. Die Abkürzung steht für „*Block Move*“: Operanden werden zusammenhängend, als Block, transferiert.

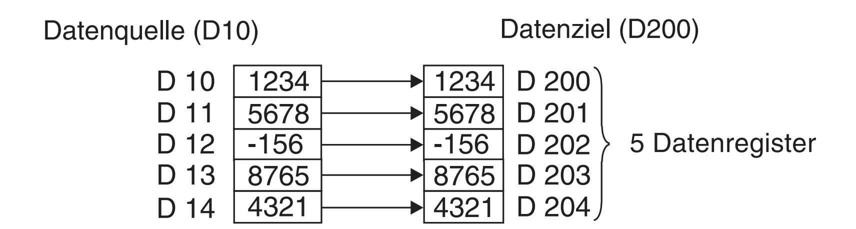
| Kontaktplan | MELSEC-Anweisungsliste | IEC-Anweisungsliste |
|---|---|---|
| BMOV_M: EN, ENO; ❶ D10 — s; ❸ 5 — n; d — D200 ❷ | BMOV D10 ❶<br>D200 ❷<br>K5 ❸ | BMOV_M D10, 5, D200<br>❶ ❸ ❷ |

❶ Datenquelle (16-Bit-Operand, angegeben wird der 1. Operand des Quellbereichs)

❷ Datenziel (16-Bit-Operand, angegeben wird der 1. Operand des Zielbereichs)

❸ Anzahl der zu übertragenen Elemente

Mit den oben angegebenen Operanden ergibt sich die folgende Funktion:




Auch eine BMOV-Anweisung kann flankengesteuert ausgeführt werden und wird in diesen Fall als BMOVP-Anweisung programmiert (siehe Abschnitt 6.2.1).

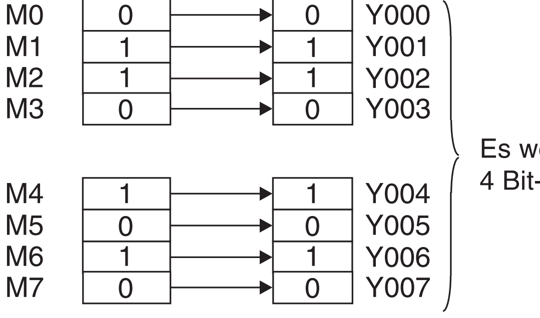
Falls mit einer BMOV-Anweisung Gruppen von Bit-Operanden transferiert werden sollen, müssen die Faktoren „K“ von Datenquelle und Datenziel identisch sein.

**Beispiel**

– Datenquelle: K1M0

– Datenziel: K1Y0

– Anzahl der zu übertragenen Elemente: 2



## 6.2.4 Transfer von gleichen Daten in mehrere Zieloperanden (FMOV)

Mit einer FMOV-Anweisung wird der Inhalt eines Wort-Operanden oder eine Konstante in mehrere, aufeinanderfolgende Wort-Operanden eingetragen. So können z. B. Datentabellen gelöscht oder Datenregister auf einen definierten Anfangswert gebracht werden.

| Kontaktplan | MELSEC-Anweisungsliste | IEC-Anweisungsliste |
|---|---|---|
| X1, FMOV_M, EN, ENO, ❶ D4 — s, d — D250 ❷, ❸ 20 — n | FMOV D4 ❶<br>D250 ❷<br>K20 ❸ | FMOV_M D4, 20, D250<br>❶ ❸ ❷ |

❶ Daten, die in die Zieloperanden eingetragen werden sollen; auch die Angabe von Konstanten ist möglich

❷ Datenziel (angegeben wird der 1. Operand des Zielbereichs)

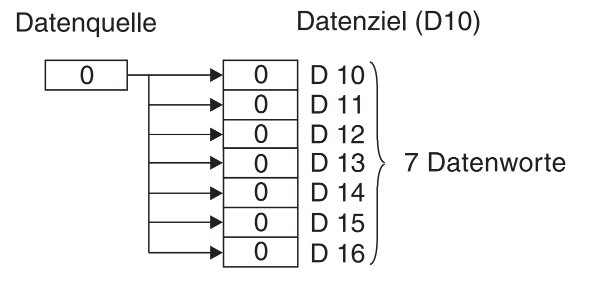
❸ Anzahl der zu beschreibenden Elemente des Zielbereichs

Im folgenden Beispiel wird in 7 Elementen der Wert „0“ eingetragen:

– Datenquelle: K0 (Konstante)

– Datenziel: D10

– Anzahl der zu beschreibenden Elemente: 7



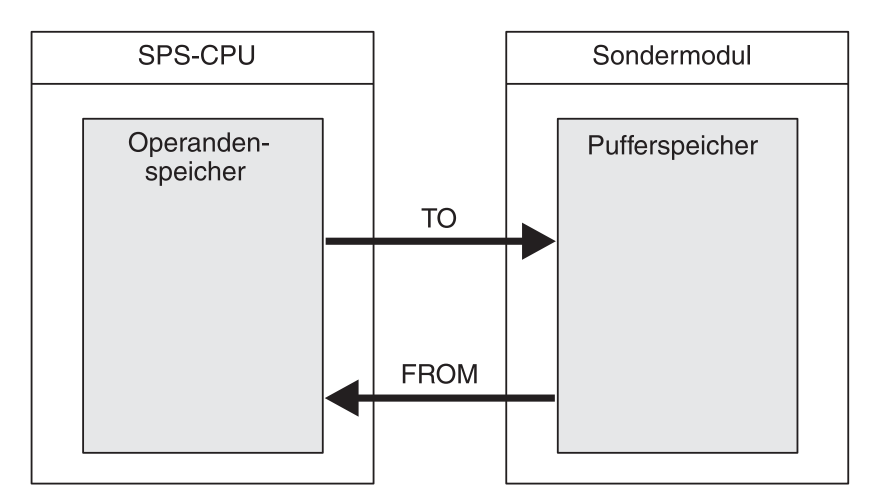

Wird statt einer FMOV-Anweisung eine FMOVP-Anweisung verwendet, erfolgt die Übertragung der Daten flankengesteuert (siehe Beschreibung der MOV-Anweisung in Abschnitt 6.2.1).

### 6.2.5 Datenaustausch mit Sondermodulen

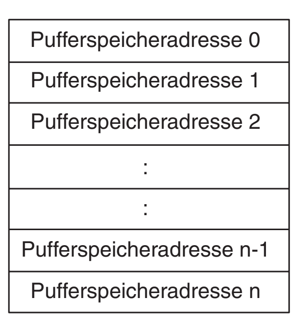
Der Funktionsumfang einer SPS des MELSEC System Q kann durch die Installation von sogenannten Sondermodulen erheblich gesteigert werden. Sondermodule erfassen zum Beispiel analoge Werte wie Ströme oder Spannungen, regeln Temperaturen oder wickeln die Kommunikation mit externen Geräten ab.

Im Sondermodul ist ein Speicherbereich eingerichtet, in dem z. B. analoge Messwerte oder empfangene Daten zwischengespeichert – gepuffert – werden. Wegen dieser Funktion wird dieser Speicherbereich als „Pufferspeicher“ bezeichnet. Auf den Pufferspeicher in einem Sondermodul kann auch die SPS-CPU zugreifen und z. B. Messwerte oder empfangene Daten lesen, aber dort auch Daten eintragen, die das Sondermodul dann weiterverarbeitet (Einstellungen für die Funktion des Sondermoduls, Sendedaten etc.).

Zusätzlich besitzen Sondermodule zum Datenaustausch mit der SPS-CPU noch digitale Ein- und Ausgänge, mit denen beispielsweise Statusmeldungen an die CPU übergeben werden. Während für die digitalen E/As der Sondermodule keine besonderen Anweisungen benötigt werden, stehen für den Datenaustausch über den Pufferspeicher eines Sondermoduls zwei Applikationsanweisungen zur Verfügung: Die FROM- und die TO-Anweisung.




Ein Pufferspeicher kann aus bis zu 32767 einzelnen Speicherzellen bestehen. Jede dieser Pufferspeicheradressen kann 16 Bit an Informationen speichern. Die Funktion einer Pufferspeicheradresse hängt von der Art des Sondermoduls ab und kann den Bedienungsanleitungen der einzelnen Sondermodule entnommen werden.




Für die korrekte Funktion benötigt eine FROM- oder TO-Anweisung bestimmte Angaben:

– Aus welchem Sondermodul sollen Daten gelesen bzw. in welches Sondermodul sollen Daten übertragen werden?
– Wie lautet die erste Pufferspeicheradresse, aus der Daten gelesen oder in die Daten eingetragen werden?
– Aus wie vielen Pufferspeicheradressen sollen Daten gelesen bzw. in wie vielen Adressen sollen Daten eingetragen werden.
– Wo in der SPS-CPU sollen die Daten aus dem Pufferspeicher abgelegt werden bzw. wo sind die Daten gespeichert, die zum Sondermodul übertragen werden sollen.

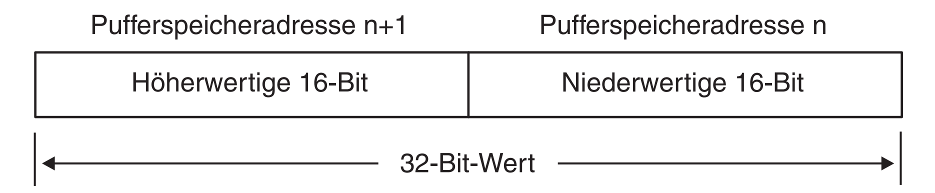
**Sondermoduladresse**

Um die Daten in das richtige Sondermodul zu transferieren oder aus dem korrekten Modul zu lesen, ist eine eindeutige Kennzeichnung der Module erforderlich. Diese ergibt sich aus dem Steckplatz des Sondermoduls auf dem Baugruppenträger bzw. den Adressbereich, den das Sondermodul mit seinen digitalen Ein- und Ausgängen belegt (siehe Abschnitt 3.2.2).

Entscheidend ist dabei die Anfangs- oder Kopfadresse des E/A-Adressbereichs. Belegt ein Sondermodul z. B. den Adressbereich X/Y010 bis Y/X01F, lautet die Anfangsadresse X/Y010. Bei der Programmierung einer FROM- oder TO-Anweisung wird die niedrigste Ziffer weggelassen und z. B. diese Adresse als „1“ angegeben. Wird durch ein Sondermodul der Adressbereich X/Y040 bis Y/X04F belegt, wird dies einer FROM- oder TO-Anweisung als „4“ mitgeteilt.

**Anfangsadresse im Pufferspeicher**

Jede der bis zu 32767 Pufferspeicheradressen kann dezimal im Bereich von 0 bis 32766 adressiert werden. 32-Bit-Daten werden so im Pufferspeicher abgelegt, dass die Speicherzelle mit der niedrigeren Adresse die niederwertigen 16-Bit und die folgende Pufferspeicheradresse die höherwertigen 16-Bit enthält.

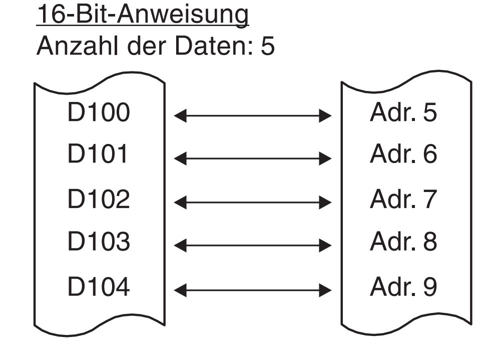



Als Anfangsadresse für 32-Bit-Daten muss daher immer die Adresse angegeben werden, welche die niederwertigen 16-Bit enthält.

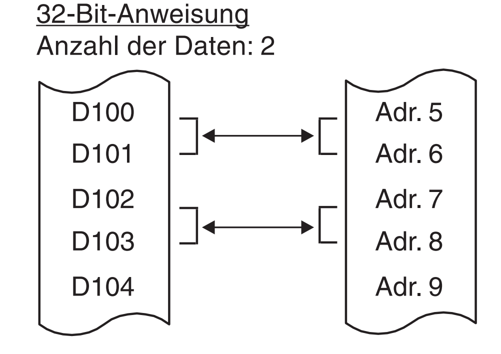
**Anzahl der zu übertragenden Daten**

Die Anzahl der Daten bezieht sich auf die zu übertragenden Dateneinheiten. Wird eine FROM- oder TO-Anweisung als 16-Bit-Anweisung ausgeführt, entspricht diese Angabe der Anzahl der Worte, die übertragen wird. Bei einer 32-Bit-Anweisung in der Form DFRO oder DTO wird die Anzahl der zu übertragenden Doppelworte angegeben.

16-Bit-Anweisung
Anzahl der Daten: 5




32-Bit-Anweisung
Anzahl der Daten: 2




**Datenziel oder -quelle in der SPS-CPU**

In den meisten Fällen werden die Daten aus Registern gelesen und an ein Sondermodul übertragen oder aus dessen Pufferspeicher in den Datenregisterbereich der SPS-CPU transferiert. Als Datenziel oder -quelle können aber auch Ausgänge und Merker oder Timer- und Counter-Istwerte dienen.

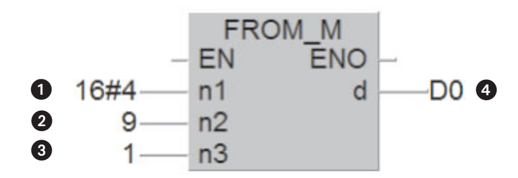
**Flankengesteuerte Ausführung der Anweisungen**

Wenn an die Abkürzung der Anweisung ein „P“ angefügt wird, erfolgt die Übertragung der Daten flankengesteuert (siehe Beschreibung der MOV-Anweisung in Abschnitt 6.2.1).

### Die FROM-Anweisung im Detail

Mit einer FROM-Anweisung werden Daten aus dem Pufferspeicher eines Sondermoduls in die SPS-CPU übertragen. Der Inhalt des Pufferspeichers ändert sich dabei nicht, die Daten werden kopiert.

Kontaktplan




MELSEC-Anweisungsliste

| | | |
|---|---|---|
| FROM | H4 | ❶ |
| | K9 | ❷ |
| | D0 | ❹ |
| | K1 | ❸ |

IEC-Anweisungsliste

FROM_M 16#4, 9 , 1 , D0
❶ ❷ ❸ ❹

❶ Kopfadresse des Sondermoduls auf dem Baugruppenträger
Die Adresse kann als dezimale oder hexadezimale Konstante (16#) angegeben werden.

❷ Anfangsadresse im Pufferspeicher
Die Angabe kann durch eine Konstante oder ein Datenregister erfolgen, das den Wert der Adresse enthält.

❸ Anzahl der zu übertragenden Daten

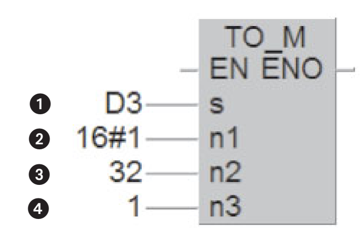
❹ Datenziel in der SPS-CPU

Im oben abgebildeten Beispiel wird aus dem Sondermodul mit der Kopfadresse X/Y040 der Inhalt der Pufferspeicheradresse 9 in das Datenregister D0 übertragen.

### Die TO-Anweisung im Detail

Mit einer TO-Anweisung werden Daten aus der SPS CPU in den Pufferspeicher eines Sondermoduls übertragen. Der Inhalt der Datenquelle wird bei diesem Kopiervorgang nicht verändert.

Kontaktplan




MELSEC-Anweisungsliste

| | | |
|---|---|---|
| TO | H1 | ❷ |
| | K32 | ❸ |
| | D3 | ❶ |
| | K1 | ❹ |

IEC-Anweisungsliste

FROM_M D3, 16#1, 32, 1
❶ ❷ ❸ ❹

❶ Datenquelle in der SPS-CPU

❷ Kopfadresse des Sondermoduls auf dem Baugruppenträger
Die Adresse kann als dezimale oder hexadezimale Konstante angegeben werden

❸ Anfangsadresse im Pufferspeicher

❹ Anzahl der zu übertragenden Daten

Im oben abgebildeten Beispiel wird der Inhalt des Datenregisters D3 in die Pufferspeicheradresse 32 des Sondermoduls mit der Kopfadresse 1 (X/Y010) übertragen.

### Direkte Adressierung des Pufferspeichers

Auf den Pufferspeicher eines Sondermoduls kann auch direkt, z. B. mit einer MOV-Anweisung, zugegriffen werden.

Das so adressierte Sondermodul kann sich auf einem Haupt- oder Erweiterungsbaugruppenträger befinden. Sondermodule in dezentralen E/A-Stationen können auf diese Weise nicht angesprochen werden.

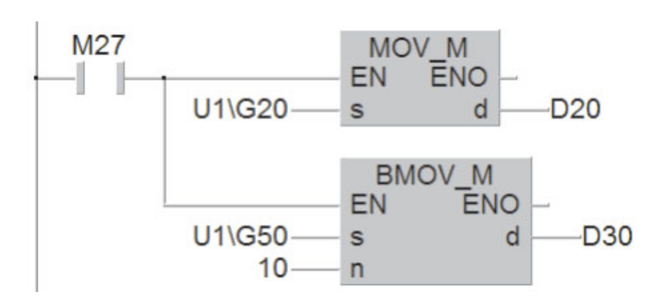
Angabe der Operandenadresse:




Bei der Operandenadresse U3\G11 zum Beispiel wird die Pufferspeicheradresse 11 im Sondermodul mit der Kopfadresse 3 (X/Y30 bis X/Y3F) angesprochen.

Wenn im folgenden Beispiel der Merker M27 gesetzt ist, wird aus dem Sondermodul mit der Kopfadresse 1 der Inhalt der Pufferspeicheradresse 20 in das Datenregister D20 kopiert. Anschließend wird mit der BMOV-Anweisung der Inhalt der Pufferspeicheradressen 50 bis 59 in die Datenregister D30 bis D39 transferiert.

Kontaktplan




MELSEC-Anweisungsliste

```
LD      M27
MOV     U1\G20
        D20
MOV     U1\G50
        D30
        K10
```

IEC-Anweisungsliste

```
LD       M27
MOV_M    U1\G20, D20
BMOV_M   U1\G50, 10, D30
```

### Automatischer Datenaustausch zwischen SPS-CPU und Sondermodul

Als Ergänzung zur Programmier-Software GX IEC Developer steht für viele Sondermodule des MELSEC System Q die optionale Konfigurations-Software GX Configurator zur Verfügung. Durch diese Software wird die Einstellung der Sondermodule vereinfacht und der Datenaustausch zwischen SPS-CPU und Sondermodul automatisiert.

Mit der Software ***GX Configurator-AD*** beispielsweise können sämtliche Einstellungen für Analogeingangsmodule vorgenommen werden. Dazu muss der Anwender nicht die Pufferspeicherstruktur des Sondermoduls kennen. Die Sondermodulparameter werden mit dem Programm in die SPS übertragen und müssen nicht mehr im SPS-Programm übergeben werden. Dadurch werden der Programmieraufwand und Fehlerquellen erheblich reduziert.

Zusätzlich kann im GX Configurator-AD angegeben werden, in welche Operanden der SPS-CPU zum Beispiel die erfassten Messwerte gespeichert werden sollen. Dieser Datentransfer läuft danach automatisch ab, FROM-/TO-Anweisungen oder der oben beschriebene direkte Pufferspeicherzugriff sind dazu nicht notwendig.

## 6.3 Vergleichsanweisungen

Um im Programm den Status von Bit-Operanden wie Eingängen oder Merkern zu prüfen, genügen die logischen Grundanweisungen, weil diese Operanden nur die beiden Zustände „0“ und „1“ annehmen können. Oft muss aber im Programm der Inhalt eines Wort-Operanden geprüft und abhängig davon eine bestimmte Aktion, wie zum Beispiel das Einschalten eines Kühlventilators beim Überschreiten einer bestimmten Temperatur, eingeleitet werden.

Dazu kann eine Ausgabeanweisung oder eine Verknüpfung von einem Vergleich abhängig gemacht werden. Neben den hier vorgestellten Vergleichsanweisungen für binäre Werte können die CPU-Module des MELSEC System Q auch Gleitkommazahlen, Binärdatenblöcke und Zeichenfolgen vergleichen.

Außer den MELSEC-Anweisungen stehen für Vergleiche auch IEC-Anweisungen zur Verfügung.

**Vergleich am Anfang einer Verknüpfung**

Kontaktplan




MELSEC-Anweisungsliste

| | | |
|---|---|---|
| ❶ | LD>= | D40 ❷ |
| | | D50 ❸ |
| | OUT | M10 |

IEC-Anweisungsliste

Diese Anweisung entspricht der Beschaltung des EN-Eingangs im Kontaktplan. „TRUE“ bedeutet, dass die Eingangsbedingung immer erfüllt ist

| | | |
|---|---|---|
| | LD | TRUE |
| ❶ | LD_GE_M | D40, D50 (❷ ❸) |
| | ST | M10 |

❶ Vergleichsbedingung

❷ Erster Vergleichswert

❸ Zweiter Vergleichswert

Ist die angegebene Bedingung erfüllt, ist der Signalzustand nach der Vergleichsanweisung gleich „1“. Der Signalzustand „0“ zeigt an, dass der Vergleich nicht erfüllt ist. In diesem Beispiel wird der Merker M10 gesetzt, wenn der Inhalt des Datenregisters D40 größer oder gleich dem Inhalt von D50 ist.

Die folgenden Vergleiche sind möglich:

– Vergleich auf „Gleich“: = (Vergleichswert 1 = Vergleichswert 2)
  IEC-Befehl: EQ (*Equal*)

  Der Ausgang der Anweisung führt nur Signalzustand „1“, wenn die Werte beider Operanden gleich groß sind.

– Vergleich auf „Größer“: > (Vergleichswert 1 > Vergleichswert 2)
  IEC-Befehl: GT (*Greater Than*)

  Der Ausgang der Anweisung führt nur Signalzustand „1“, wenn der 1. Vergleichswert größer als der 2. Vergleichswert ist.

– Vergleich auf „Kleiner“: < (Vergleichswert 1 < Vergleichswert 2)
  IEC-Befehl: LE (*Less Than*)

  Der Ausgang der Anweisung führt nur Signalzustand „1“, wenn der 1. Vergleichswert kleiner als der 2. Vergleichswert ist.

– Vergleich auf „Ungleich“: <> (Vergleichswert 1 ungleich Vergleichswert 2)
  IEC-Befehl: NE (*Not Equal*)

Der Ausgang der Anweisung führt nur Signalzustand „1“, wenn der 1. und der 2. Vergleichswert ungleich sind.

– Vergleich auf „Kleiner-Gleich“: **<=** (Vergleichswert 1 ≤ Vergleichswert 2)
IEC-Befehl: LE (*Less Equal*)

Der Ausgang der Anweisung führt Signalzustand „1“, wenn der 1. Vergleichswert kleiner oder gleich dem 2. Vergleichswert ist.

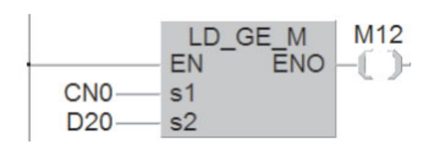
– Vergleich auf „Größer-Gleich“: **>=** (Vergleichswert 1 ≥ Vergleichswert 2)
IEC-Befehl: GE (*Greater Equal*)

Der Ausgang der Anweisung führt Signalzustand „1“, wenn der 1. Vergleichswert größer oder gleich dem 2. Vergleichswert ist.

Wenn 32-Bit-Daten verglichen werden sollen, muss der Anweisung ein „D“ (für „Doppelworte“ hinzugefügt werden (Zum Beispiel LD**D**_EQ-M oder LD**D**_GE_M)

**Beispiele für Vergleiche am Anfang von Verknüpfungen**

Kontaktplan




MELSEC-Anweisungsliste

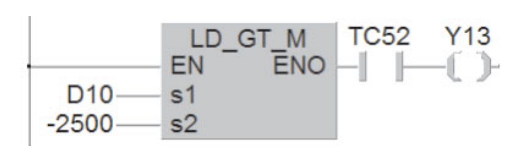
| | |
|---|---|
| LD>= | C0 |
| | D50 |
| OUT | M12 |

IEC-Anweisungsliste

| | |
|---|---|
| LD | TRUE |
| LD_GE_M | CN0, D20 |
| ST | M12 |

Der Merker M12 hat den Signalzustand „1“, wenn der Zählerstand von C0 dem Inhalt von D20 entspricht oder größer ist.

Kontaktplan




MELSEC-Anweisungsliste

| | |
|---|---|
| LD> | D10 |
| | K-2500 |
| AND | T52 |
| OUT | Y13 |

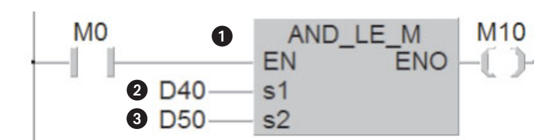
IEC-Anweisungsliste

| | |
|---|---|
| LD | TRUE |
| LD_GT_M | D10, -2500 |
| AND | TC52 |
| ST | Y13 |

Wenn der Inhalt von D10 größer als -2500 und der Timer T52 abgelaufen ist, wird der Ausgang Y13 eingeschaltet.

### Vergleich als UND-Verknüpfung

Kontaktplan




MELSEC-Anweisungsliste

| | | |
|---|---|---|
| | LD | M0 |
| ❶ | AND<= | D40 ❷ |
| | | D50 ❸ |
| | OUT | M10 |

IEC-Anweisungsliste

| | | |
|---|---|---|
| | LD | M0 |
| ❶ | AND_GE_M | D40, D50 |
| | | ❷ ❸ |
| | ST | M10 |

❶ Vergleichsbedingung

❷ Erster Vergleichswert

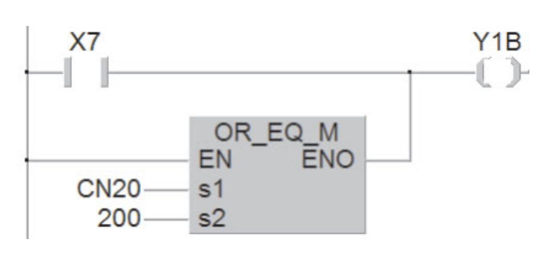
❸ Zweiter Vergleichswert

Ein UND-verknüpfter Vergleich kann im Programm wie eine normale AND-Anweisung verwendet werden (siehe Kap. 4).

Die Vergleichsmöglichkeiten entsprechen denen der oben beschriebenen Vergleiche am Anfang einer Verknüpfung. Im oben gezeigten Beispiel wird M10 gesetzt, wenn M0 „1“ ist **und** der Inhalt von D40 kleiner oder gleich dem Inhalt von D50 ist.

### Vergleich als ODER-Verknüpfung

Kontaktplan




MELSEC-Anweisungsliste

| | | |
|---|---|---|
| | LD | X7 |
| ❶ | OR= | C20 ❷ |
| | | K200 ❸ |
| | OUT | Y1B |

IEC-Anweisungsliste

| | | |
|---|---|---|
| | LD | X7 |
| ❶ | OR_EQ_M | CN20, 200 |
| | | ❷ ❸ |
| | ST | Y1B |

❶ Vergleichsbedingung

❷ Erster Vergleichswert

❸ Zweiter Vergleichswert

Im Programm kann ein ODER-verknüpfter Vergleich wie eine OR-Anweisung verwendet werden (siehe Kap. 4). In diesem Beispiel wird der Ausgang Y1B eingeschaltet, wenn der Eingang X7 eingeschaltet ist **oder** der Counter C20 den Istwert „200“ erreicht hat.

# 6.4 Arithmetische Anweisungen

Alle CPU-Module des MELSEC System Q beherrschen die vier Grundrechenarten und können addieren, subtrahieren, multiplizieren und dividieren. Es stehen MELSEC-Anweisungen für arithmetische Operationen mit Binärwerten, Gleitkommazahlen, BCD-Daten, Zeichenfolgen und Binärdatenblöcken zur Verfügung.

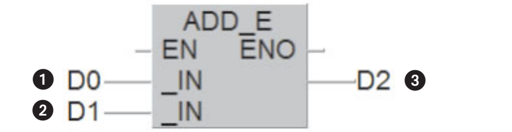
Zusätzlich können für die Programmierung im Kontaktplan oder der IEC-Anweisungsliste beim GX IEC Developer IEC-Anweisungen genutzt werden. In diesem Abschnitt werden nur diese IEC-Anweisungen beschrieben. Eine ausführliche Beschreibung der MELSEC-Anweisungen finden Sie in der Programmieranleitung zur MELSEC A/Q-Serie und dem MELSEC System Q (Artikel-Nr. 87 432).

Die IEC-Anweisungen zur Addition, Subtraktion, Multiplikation und Division können für die Datentypen INT (ganzzahlige 16-Bit-Daten), DINT (ganzzahlige 32-Bit-Daten) und REAL (Gleitkommazahlen) verwendet werden. Operanden vom Typ DINT und REAL können nicht direkt mit der Anweisung eingegeben werden und müssen als Variable deklariert werden (siehe Abschnitt 4.6.2).

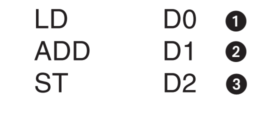
## 6.4.1 Addition

Mit einer ADD-Anweisung werden Werte addiert und das Ergebnis gespeichert.

Kontaktplan




IEC Anweisungsliste


```
LD    D0  ❶
ADD   D1  ❷
ST    D2  ❸
```

❶ Erster Quelloperand oder Konstante

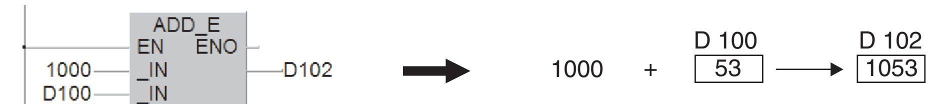
❷ Zweiter Quelloperand oder Konstante

❸ Operand, in dem das Ergebnis der Addition eingetragen wird

Im oben abgebildeten Beispiel werden bei Ausführung der ADD-Anweisung die Inhalte der Datenregister D0 und D1 addiert und das Ergebnis in D2 gespeichert.

**Beispiele**

Zum Inhalt des Datenregisters D100 wird der Wert „1000“ addiert:




Das Ergebnis kann auch wieder in einem der Quelloperanden eingetragen werden. Beachten Sie aber bitte, dass sich das Ergebnis in jedem Programmzyklus ändert, wenn die ADD-Anweisung zyklisch ausgeführt wird. Dem kann vorgebeugt werden, indem die Addition flankengesteuert ausgeführt wird.

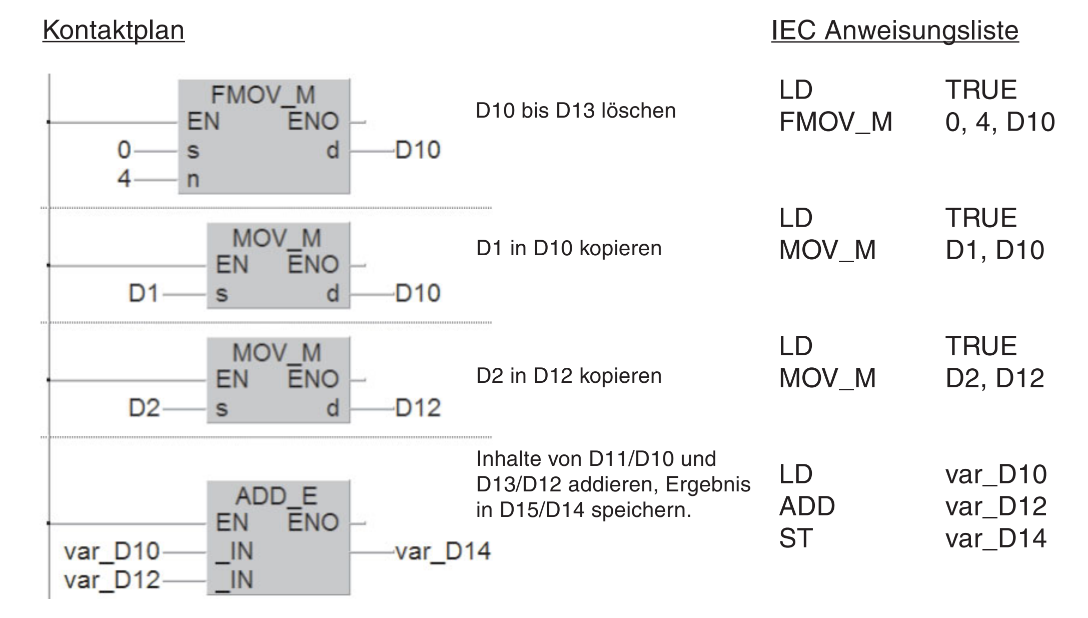
ADD_E: EN, ENO; D0 — _IN, 25 — _IN → D0

D 0 [18] + 25 → D 0 [43]

Bei der Addition werden die Vorzeichen der Werte berücksichtigt (z. B. 10 + (-5) = 5).

Bei der ADD-Anweisung müssen die Ein- und Ausgangsvariablen vom gleichen Datentyp sein. Das kann zu Problemen führen, wenn das Ergebnis der Addition den Wertebereich der Variablen überschreitet. Werden beispielsweise die zwei 16-Bit-Festpunktzahlen „32700“ und „100“ addiert, wird nicht, wie zu erwarten, „32800“ als Ergebnis gespeichert, sondern „-32736“, weil eine 16-Bit-Variable nur den maximalen Wert „32767“ darstellen kann. Ein Überlauf wird als negative Zahl interpretiert und führt dadurch zu einem falschen Ergebnis.

Eine Lösungsmöglichkeit ist, die zu addierenden Werte vor der Addition in 32-Bit-Variable zu kopieren und anschließend die Addition mit 32-Bit-Variablen auszuführen.

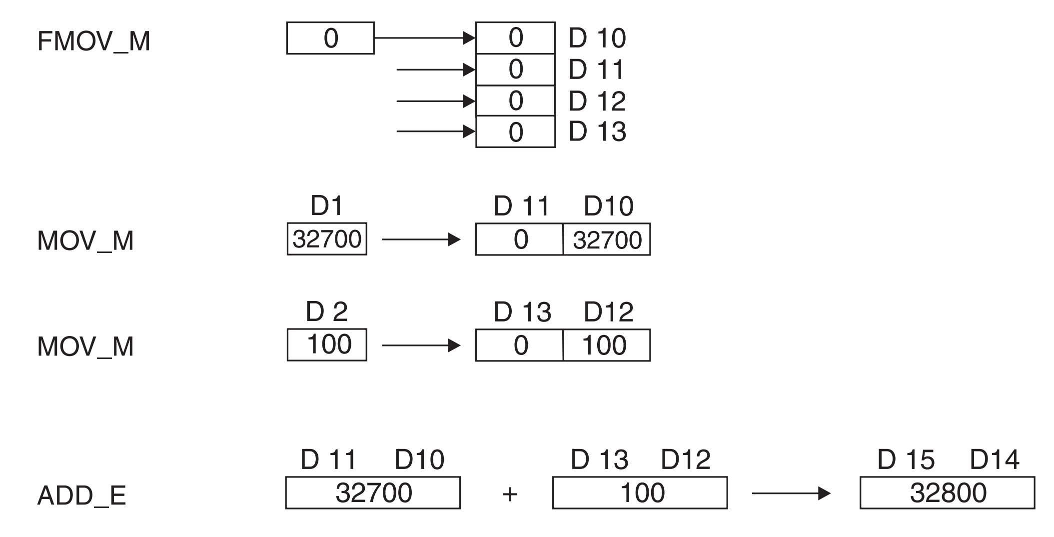



Da 32-Bit-Variablen nicht direkt mit der ADD-Anweisung angegeben werden können, ist eine Definition als Globale Variable erforderlich:

| | Klasse | Bezeichner | MIT-Adr. | IEC-Adress | Typ |
|---|---|---|---|---|---|
| 0 | VAR_GLOBAL | var_D10 | D10 | %MD0.10 | DINT |
| 1 | VAR_GLOBAL | var_D12 | D12 | %MD0.12 | DINT |
| 2 | VAR_GLOBAL | var_D14 | D14 | %MD0.14 | DINT |

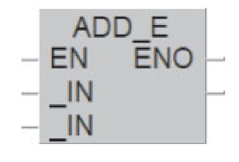
Die Name der Variablen (*Bezeichner*) können frei gewählt werden. Zum besseren Verständnis wurden hier die Operandenadressen übernommen.

Mit den Zahlenwerten von oben werden die Inhalte der Datenregister bei der Ausführung dieser vier Anweisungen wie folgt verändert:

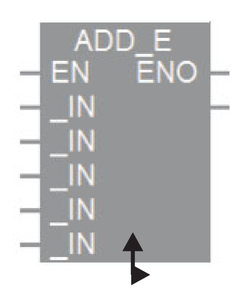



Doppelregister D14 enthält das korrekte Ergebnis der Addition.

Die ADD-Anweisung ist nicht auf zwei Eingangsvariable beschränkt. Bis zu 28 Eingangsvariable können angegeben werden. Bei der Kontaktplanprogrammierung geht das so:




Wählen Sie im Dialogfenster „Auswahl Funktionsblock“ (siehe Abschnitt 4.7.7) die ADD_E-Anweisung und positionieren Sie sie in den Editierbereich.




Klicken Sie dann in die Anweisung, die dadurch eine andere Farbe annimmt und bewegen Sie den Cursor nach unten, bis er sich in einen Doppelpfeil verwandelt.

Betätigen Sie dann die linke Maustaste und führen Sie bei immer noch betätigter Maustaste den Cursor weiter nach unten, bis die gewünschte Anzahl Eingangsvariablen angezeigt wird.

Bei der Programmierung in IEC-Anweisungsliste geben Sie einfach die ADD-Anweisung mehrmals nacheinander ein. Zum Beispiel:



## 6.4.2 Subtraktion

Zur Subtraktion von zwei numerischen Werten (Inhalte von 16- oder 32-Bit-Operanden oder Konstanten) kann die SUB-Anweisung verwendet werden. Das Ergebnis der Subtraktion wird in einem dritten Operanden abgelegt.

Kontaktplan




IEC Anweisungsliste

| | | |
|---|---|---|
| LD | D0 | ❶ |
| SUB | D1 | ❷ |
| ST | D2 | ❸ |

❶ Minuend (Von diesen Wert wird abgezogen)

❷ Subtrahend (Dieser Wert wird abgezogen.)

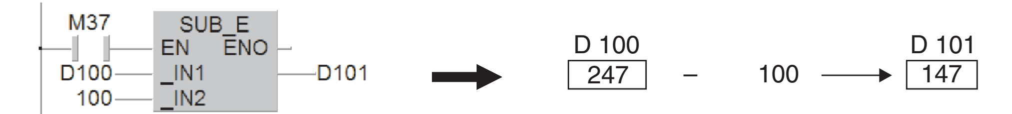
❸ Differenz (Ergebnis der Subtraktion)

Bei der SUB-Anweisung müssen die Ein- und Ausgangsvariablen vom gleichen Datentyp sein.

Mit der oben abgebildeten Anweisung wird der Inhalt von D1 vom Inhalt von D0 subtrahiert und das Ergebnis in D2 gespeichert.

**Beispiele**

Wenn der Merker M37 gesetzt ist , wird vom Inhalt des Datenregisters D100 der Wert „100“ subtrahiert und das Ergebnis in D101 gespeichert:




Die Werte werden unter Berücksichtigung der Vorzeichen subtrahiert:

SUB_E
EN ENO
D10 _IN1
D11 _IN2
D12

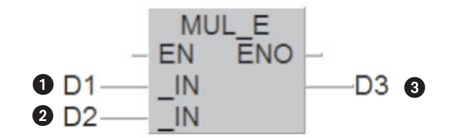
D 10: 5 – D 11: -8 → D 12: 13

Wie bei der ADD-Anweisung kann das Ergebnis kann auch wieder in einen der Quelloperanden eingetragen werden. Wird die SUB-Anweisung zyklisch ausgeführt, ändert sich der Inhalt dieses Operanden in jedem Programmzyklus.

## 6.4.3 Multiplikation

Durch eine MUL-Anweisung multipliziert die SPS-CPU 16- oder 32-Bit-Werte und speichert das Ergebnis.

Kontaktplan





IEC Anweisungsliste




❶ Multiplikand

❷ Multiplikator

❸ Produkt (Multiplikand x Multiplikator = Produkt)

Im oben dargestellten Beispiel werden bei Ausführung der MUL-Anweisung die Inhalte der Datenregister D1 und D2 multipliziert und das Ergebnis in D3 gespeichert.

**HINWEIS**

Bei der MUL-Anweisung müssen die Ein- und Ausgangsvariablen vom gleichen Datentyp sein. Wenn das Multiplikationsergebnis größer ist als der Wert, der als Maximalwert in einer 16- oder 32-Bit-Variable dargestellt werden kann, gehen die oberen Bit verloren, und das Produkt wird nicht korrekt dargestellt. Sollen 16-Bit-Werte multipliziert werden, können, wie bei der ADD-Anweisung im Abschnitt 6.4.1 beschrieben, die Werte vorher in 32-Bit-Variable kopiert werden. Die MUL-Anweisung wird dann ebenfalls mit 32-Bit-Operanden ausgeführt und das Ergebnis korrekt ausgegeben.

Eine MUL-Anweisung kann bis zu 28 Eingangsvariablen haben. Die Einstellung wird so vorgenommen wie bei der ADD-Anweisung (siehe Abschnitt 6.4.1).

**Beispiele**

Multiplikation der Inhalte von D1 und D2 und Speicherung des Ergebnisses in D3:

MUL_E: EN ENO; D1 — _IN; D2 — _IN; — D3 ➡ D 1 [144] x D2 [17] → D 3 [2448]

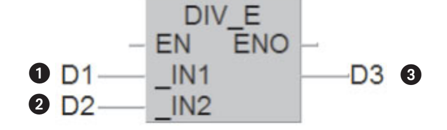
Die Multiplikation erfolgt unter Berücksichtigung der Vorzeichen. In diesem Beispiel wird der Inhalt von D10 mit der Konstanten „-5“ multipliziert:

MUL_E: EN ENO; D10 — _IN; -5 — _IN; — D20 ➡ D 10 [8] x -5 → D 20 [-40]

## 6.4.4 Division

Um zwei Zahlen zu dividieren, kann die DIV-Anweisung verwendet werden.

Kontaktplan




IEC Anweisungsliste

| | | |
|---|---|---|
| LD | D1 | ❶ |
| DIV | D2 | ❷ |
| ST | D3 | ❸ |

❶ Dividend

❷ Divisor

❸ Quotient (Ergebnis der Division: Dividend ÷ Divisor = Quotient)

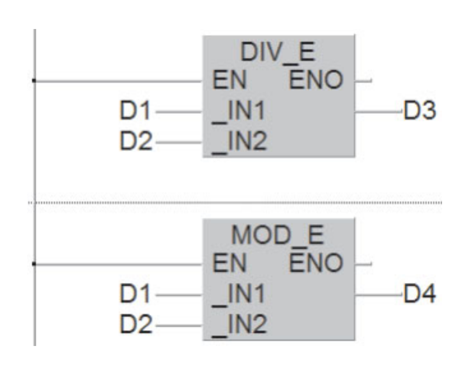
In diesem Beispiel wird der Inhalt des Datenregisters D1 durch den Inhalt von D2 geteilt und das Ergebnis in D3 gespeichert.

**HINWEIS**

Der Divisor darf nicht den Wert „0“ annehmen. Eine Division durch „0“ ist nicht möglich und führt zu einem Fehler, der die SPS-CPU stoppt. (Dieser Fall kann zum Beispiel auftreten, wenn, wie im oben gezeigten Beispiel, die Division mit den Inhalten von Datenregistern ausgeführt wird und die Register nach einem Reset gelöscht sind. Um einen Stopp der SPS zu vermeiden, könnte im SPS-Programm das Datenregister mit den Divisor **vor** der Ausführung der DIV-Anweisung auf einen definierten Wert gesetzt werden.)

Die Ein- und Ausgangsvariablen der DIV-Anweisung müssen vom gleichen Datentyp sein. Bei der Division von Festzahlen (INT oder DINT) wird als Quotient nur das ganzzahlige Ergebnis gespeichert. Der nicht teilbare Rest kann mit einer MOD-Anweisung ermittelt werden.

Kontaktplan




IEC Anweisungsliste

| | |
|---|---|
| LD | D1 |
| DIV | D2 |
| ST | D3 |
| LD | D1 |
| MOD | D2 |
| ST | D4 |

Die MOD-Anweisung wird mit den selben Eingangsvariablen „versorgt“ wie die DIV-Anweisung. Im Beispiel oben wird der Inhalt von D1 wird durch den Inhalt von D2 geteilt und das Ergebnis in D3 und der Rest in D4 gespeichert:

D 1: 40 ÷ D2: 6 → D3: 6 Quotient (6 x 6 = 36) (Ausgang der DIV-Anweisung)

D 4: 4 Rest (40 - 36 = 4) (Ausgang der MOD-Anweisung)

Bei der Division werden die Vorzeichen berücksichtigt. Im folgenden Beispiel wird der Zählerstand von C0 durch den Inhalt von D10 geteilt:

DIV_E: EN ENO; CN0 — _IN1; D10 — _IN2; — D200

➡ C 0: 36 ÷ D 10: -5 → D 200: -7

### 6.4.5 Kombination von arithmetischen Anweisungen

In der Praxis kommt man selten mit nur einer Berechnung aus. Zur Lösung komplexer Aufgaben können arithmetische Anweisungen sehr einfach kombiniert werden.

Die Addition der Inhalte der Datenregister D101, D102, die anschließende Multiplikation mit dem Faktor „4“ und schließlich die Division durch „9“ könnte zum Beispiel so realisiert werden:

Kontaktplan

M101
D101
D102
ADD_E
EN ENO
_IN
_IN
4
MUL_E
EN ENO
_IN
_IN
9
DIV_E
EN ENO
_IN1
_IN2
D103

# Index

## D



## E



## F



## G



## H



## I



## K



## L



## M

## N



## O



## P



## Q



## R



## S